



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación :

INGENIERO EN INFORMÁTICA

Título del proyecto:

TOPOLOGÍA, MODELADO Y RIGGING FACIAL

Pablo J. Pinilla Sánchez

Iosu Azkue Odriozola

Pamplona, Diciembre 2013

Índice

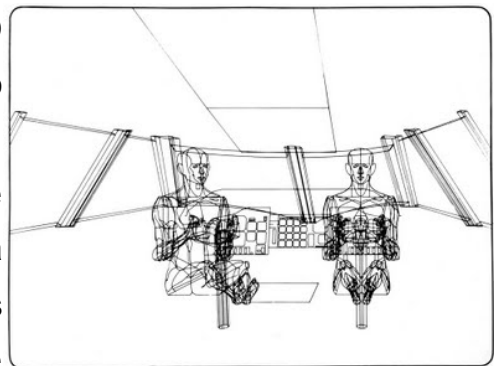
1.	Introducción.....	4
1.1	Historia.....	4
1.2	En qué consiste.....	5
1.3	Propósito del proyecto.....	7
2.	Modelado.....	9
2.1	Introducción.....	9
2.2	Técnicas generales de modelado.....	9
2.2.1.	Procedural Modeling.....	10
2.2.2.	3D Scanning.....	10
2.2.3.	NURBS/ Spline Modeling	11
2.2.4.	Digital Sculpting.....	12
2.2.5.	Displacement.....	12
2.3	Técnicas poligonales.....	13
2.3.1.	Primeros conceptos.....	13
2.3.2.	Teoría geométrica y polígonos.....	14
2.3.3.	Construcción de mallas poligonales.....	14
2.3.3.1.	Primitive Modeling.....	14
2.3.3.2.	Spline Extrusion.....	15
2.3.3.3.	Box/ Subdivision Modeling.....	16
2.3.3.4.	Edge/ Contour Modeling	17
2.3.4.	Subdivision Surface, explicación teórica.....	18
2.3.4.1.	Ventajas y desventajas.....	19
2.4	Topología.....	20
2.4.1.	Tipos generales.....	20
2.4.1.1.	High Polycount.....	21
2.4.1.2.	Classic (Limited) Game.....	23
2.4.1.3.	Animación.....	24
2.4.1.4.	Juegos Modernos.....	26
2.4.1.5.	Strict Quad.....	29
2.4.1.6.	Box Topology.....	31
2.4.2.	Requisitos mínimos.....	33
2.4.3.	¿Por qué es importante la topología?.....	35
2.4.4.	Conceptos básicos.....	36

2.4.5.	Stepping.....	42
2.4.6.	Edgeloops.....	49
2.5	Modelado del rostro.....	53
2.5.1.	Topología.....	55
2.5.2.	Pre-modelado.....	59
2.5.3.	Box Modeling.....	62
2.5.4.	Edge Modeling	66
2.5.5.	Creación de la oreja.....	69
2.5.6.	Modelado final.....	71
2.5.7.	Conclusión final.....	76
3.	Rigging.....	77
3.1	Introducción: Morpher vs Bones.....	77
3.2	Rigging de la cabeza.....	81
4.	Bibliografía.....	95

1. INTRODUCCIÓN

1.1 Historia

Desde los años 60 se viene desarrollando una rama de la tecnología centrada en los gráficos producidos por ordenador. Desde 1957 con la creación de la primera imagen producida por escanear una foto hasta las nuevas películas actuales de ordenador han pasado muchos hitos, desde acuñar el término de “computer graphics”, la creación del primer cuerpo humano (Boeing man), la primera Interfaz Gráfica, se fueron creando poco a poco los primeros lenguajes y programas para el modelado y la animación y creando nuevas herramientas. Hasta 1978 no se usó el 3D como recurso para películas, en Futureworld, sacando por pantalla la animación de una cara y de una mano. Desde entonces se ha usado para todo tipo de películas, desde su uso como soporte para determinadas secuencias, hasta películas completas de animación. También cabe destacar la creación de estudios como Pixar (1986). A su vez a partir de los años 80 se empezaron a crear las compañías que desarrollan software específico para la creación de animación 3D, tales como Autodesk, Softimage... A partir de los 90 su uso se empezó a generalizar en cine y televisión, sobre todo para el uso de publicidad y de efectos especiales, como control de masas (usado en películas tan variadas como Batman Returns para la simulación de los murciélagos como en El Rey León para la escena de la estampida). Se desarrollaron técnicas de captura de movimientos y de cámara y evoluciona día a día, siendo una ciencia en plena expansión.



1.2 En qué consiste

Los gráficos 3d generados por ordenador son gráficos que usan una representación tridimensional de data geométrico que es almacenado en el ordenador para poder calcularlo y renderizarlo a imágenes de 2 dimensiones. Estas imágenes son almacenadas para poder verlas después o en tiempo real.

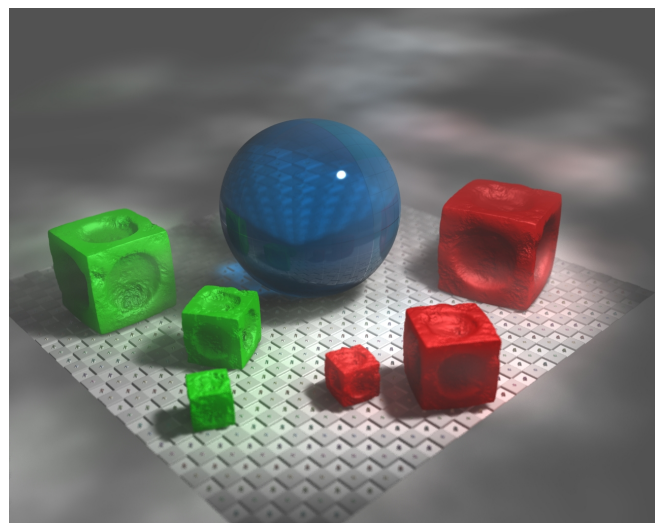
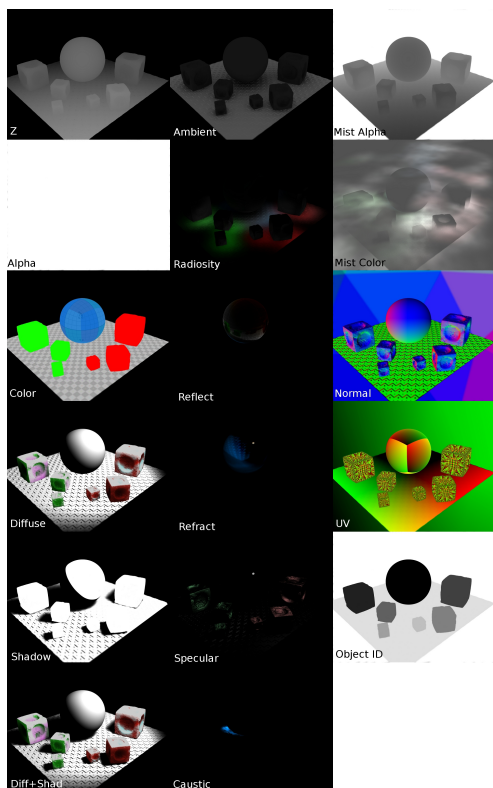
El proceso para producir estas imágenes sigue una serie de pasos, aunque depende del uso que se les quieran dar a esas imágenes. Por lo general, todos los usos se componen de tres fases principales, pre-producción, producción y post-producción.

La primera fase, la pre-producción, consiste en preparar todas las herramientas, conceptos y tener toda la planificación de todo lo que se va a hacer posteriormente. Vamos a exponer aquí el flujo de trabajo de un corto o de una animación, siendo extrapolable para cualquier otro proceso como la creación de material 3d para videojuegos o para publicidad, o una infografía de arquitectura o interiorismo, para poder ilustrar todo el flujo. Primeramente se juntan las ideas que se quieren exponer en el corto, qué se quiere obtener y cómo. Tras esto se guioniza todo lo que va a ocurrir en el corto. Se realizan los diferentes concept art de todos los elementos de la escena, y si existen personajes, se diseñan y se muestran en diferentes poses. Además se diseña el ambiente en el que se desarrollará la escena. Con el guión y todo lo anterior se realiza un storyboard, que es un conjunto de dibujos que forman la secuencia que servirá como guía según el guión realizado y muestra planos que se van a hacer y todo lo necesario y con él se crea la última fase de la pre-producción.



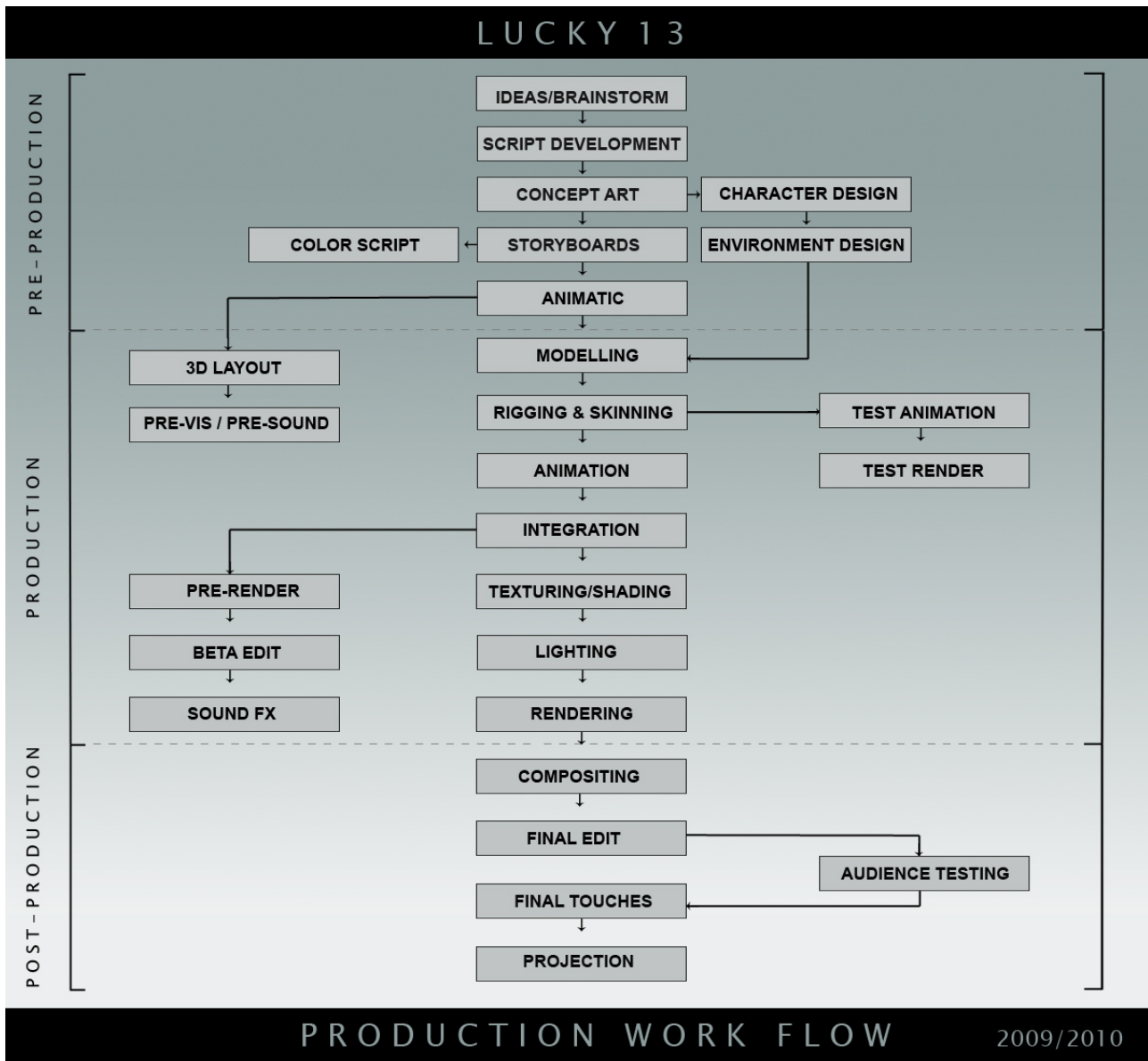
Con todo esto ya en marcha, se empieza la etapa de producción, que es en la que nos vamos a centrar en este proyecto, sobre todo en sus dos primeras fases. La primera etapa es la de modelado, consistente en la creación del modelo 3d, que básicamente es en el

proceso de dar forma a un objeto. Para esto hay dos formas, a través de modelos que crea desde cero el artista o ingeniero, y modelos escaneados a través de un escáner. Además de las dos formas típicas, se pueden modelar objetos a través de simulación física y procedimentalmente. Tras haber modelado todos los objetos de la escena, se colocan según sus propiedades espaciales de cada uno de ellos, respetando las proporciones y situándolos en la imagen. Tras la fase de modelado, hace falta riggear los diferentes personajes y elementos que hay que animar en la escena. Riggear significa introducir huesos en los modelos, que nos facilitarán la tarea de posicionar cada parte de los modelos, así como su posterior animación. La siguiente etapa es la animación de todos los elementos y huesos que hemos metido en la escena para conseguir el resultado requerido. Después se añaden todos los procesos de partículas que se necesiten meter para realizar la simulación. Posteriormente se texturiza todos los elementos para darles los colores y efectos deseados y se ilumina la escena con la tonalidad que se quiera transmitir. Por último se renderiza, que es calcular todos los elementos que se han plasmado en las fases anteriores y conseguir la imagen final.



La última fase es el post-procesado, que es la parte en la que se montan y se acaba la animación final. Tiene varias etapas, empezando por la composición. Al renderizar de un

modo profesional se producen varias imágenes de un mismo fotograma, en unas realizando determinadas luces, otras enfatizando las sombras, otras con efectos generados adicionalmente, y todo ello se combinan en una misma imagen final. Después se hace como en otras producciones, se edita todo para sacar las escenas que no resultan y se dan los últimos retoques antes de la exposición final.



1.3 Propósito del proyecto

En este proyecto se tratará de centrarse en las dos primeras fases de la producción, modelado y rigging y animación, centrándonos en una de las partes más complicadas de

modelado y animación, que es el rostro humano. Primeramente se centrará en qué consiste el modelado, y buenas técnicas, para posteriormente centrarnos en cómo modelar bien una cabeza humana. Para ello debemos saber en qué consiste una buena topología y como conseguirla.

Después tocará animar esa cabeza, y para eso buscaremos las diferentes posibilidades que se nos ofrecen y veremos sus diferentes virtudes y contras, intentando sacar siempre las formas más efectivas de hacerlo.

2. Modelado

2.1 Introducción

Se llama modelado 3D al proceso de desarrollo de una representación matemática de una superficie tridimensional de un objeto a través de software especializado. El producto de este desarrollo se llama modelo 3D. Los modelos 3d representan a un objeto usando un conjunto de puntos en el espacio, conectado por triángulos, líneas, superficies curvas y más. Los modelos pueden ser creados automáticamente o manualmente.

Hoy en día son usados en gran variedad de ámbitos. La industria médica los usa con gran precisión para representar órganos humanos, el cine los usa como personajes y objetos para películas animadas, la industria química para hacer modelos de gran detalle de sus partículas, y tenemos otras ramas como arquitectura, videojuegos, ingeniería o geología que los usan como algo estándar.

Los modelos pueden dividirse primeramente en dos categorías:

***Sólidos:** Estos modelos definen el volumen del objeto al que representan. Son mucho más realistas y difíciles de construir. Son usados en simulaciones médicas y de ingeniería, o aplicaciones visuales especializadas como ray tracing.

***Shell:** Estos modelos representan la superficie, los bordes del objeto pero no el volumen. Se asemejan a una cáscara de huevo. Son mucho más fáciles de crear y modificar y la gran mayoría de modelos suelen ser de este tipo.

Centrándonos en este último tipo, cabe destacar cómo se representan en estos programas gráficos. Se representan generalmente con mallas poligonales, que se refinan con superficies subdivididas, generadas a través de procesos iterativos; aunque no sólo se utiliza la técnica de malla sino que a veces existan representaciones con puntos. Por último se usan a veces representaciones de nivel para cambios topológicos como por ejemplo los fluidos.

2.2 Técnicas generales de modelado

Para poder crear un modelo, hay diferentes vías por las que se puede desarrollar, cada cual orientada para un tipo diferente:

2.2.1 Procedural Modeling

La palabra *procedural*, en gráficos por computador, se refiere a cualquier elemento generado algorítmicamente, en vez de haber sido creado por las manos de un artista. Aquí las escenas u objetos son creados por reglas definidas o parámetros.

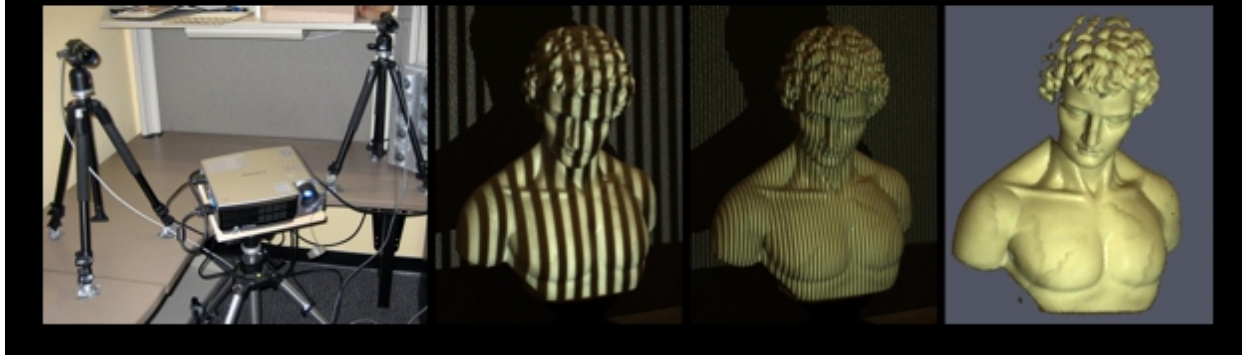
Por todo ello se utiliza masivamente en la creación de paisajes, desde desiertos a montañosos, pasando por la costa, ya que se pueden introducir parámetros como el follaje, la densidad y la elevación. También se pueden combinar otras técnicas como fractales o algoritmos recursivos para la creación de arbustos y árboles, ya que con un par de parámetros se pueden modificar aspectos que serían muy costosos de modificar a mano. Por último se usan en la creación de paisajes de ciudad.



2.2.2 3D Scanning

3D Scanning es un método de digitalizar objetos del mundo real cuando es necesario un nivel realmente alto de fotorealismo. Se coge un objeto del mundo real (o un actor) y es escaneado, analizado y los datos (normalmente una nube de puntos x, y, z) es usada para generar una malla precisa o una superficie NURBS. A pesar de que se usan para muchos y diferentes objetos, hay otros muchos que se necesita aún la mano de un artista para crearlos porque no existen en la vida real, por lo que esta técnica no puede ser usada siempre.

3D Scanning using Structured Lighting



2.2.3 NURBS/ Spline Modeling

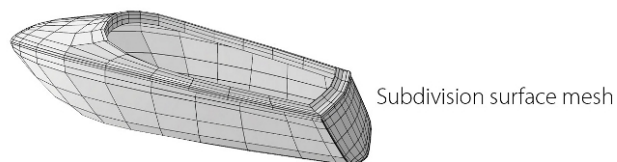
NURBS (Non-Uniform Rational B-Splines) es uno de los tres tipos de geometría que se usan en los paquetes de 3d, junto con la geometría poligonal y las *subdivision surfaces*.

*Non-Uniform se refiere a grado de parametrización de la curva. Las curvas no uniformes permiten entre otras cosas, la presencia de multi-knots (existencia de varios puntos editables en un mismo punto físico del espacio) que son necesarios para la representación de las curvas Bézier.

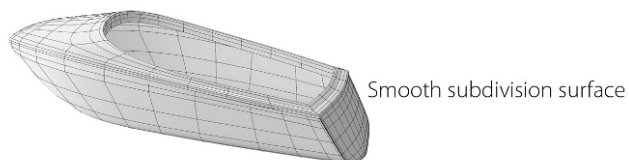
*Rational se refiere al trasfondo matemático de la representación. Esta propiedad permite representar exactamente curvas de todo tipo, desde las parabólicas, círculos y elipses, a las curvas de estilo libre.

*B-Splines son las curvas polinomiales que se usan y tienen una representación paramétrica.

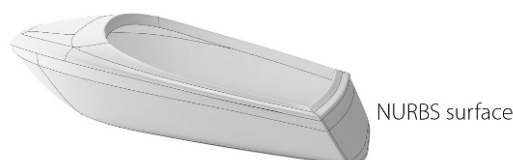
Las superficies NURBS se usan en bastantes sitios gracias al grado de suavidad que alcanzan las curvas, y se usan sobre todo en el diseño industrial, o visualizaciones científicas.



Subdivision surface mesh



Smooth subdivision surface



NURBS surface

2.2.4 Digital Sculpting

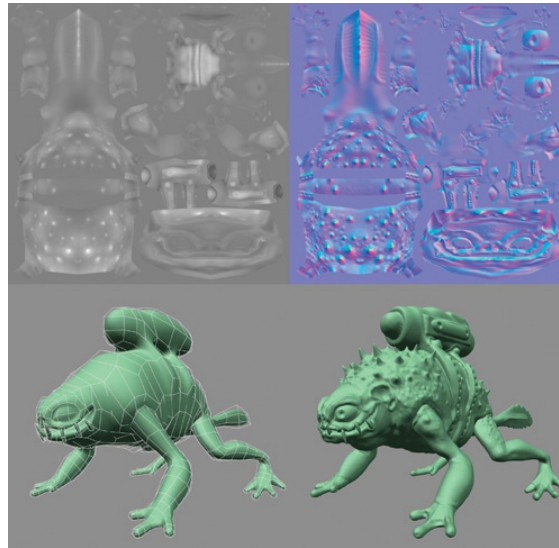
Es una técnica revolucionaria que permite a los modeladores olvidarse de las reglas de topología y flow edge y crear intuitivamente los modelos como si estuviesen esculpiendo en arcilla. Las mallas son creadas orgánicamente a través de una tableta gráfica, y modelan el objeto tal y como lo haría un escultor tradicional con sus paletas y trozos de arcilla. Esto hace que el proceso sea mucho más rápido, más eficiente y permiten a los artistas trabajar con mallas de alta resolución que contienen millones de polígonos. Claro está que para trabajar con estos modelos, hace falta software especializado como Zbrush (el más importante y el que sirve de estándar en la industria), Mudbox o Sculptrix, y estas mallas en el resto de programas son realmente difíciles de mover, por lo que no son muy usadas en animación a no ser que contemos con granjas de equipos para poder usarlos. A pesar de todo, estos modelos son conocidos por sus niveles de detalle mucho más naturales a la realidad.



2.2.5 Displacement

Técnica usada para añadir detalles a personajes y otros objetos, es bastante común en los paquetes de modelado. Consiste en añadir detalles a una malla ya creada a través de mapas de desplazamiento, que son mapas en escala de grises. Estos mapas dan un valor de altura según el nivel de gris que dispongamos. Es muy usada en la creación de

paisajes y objetos similares. Últimamente se está desechando ya que no es muy eficiente.



2.3 Técnicas poligonales

Técnicas basadas en la creación de polígonos. Son las más comunes y usadas, y las nuevas técnicas están basadas en ellas. Están basadas en primeras instancias en técnicas de *low poly* (baja resolución).

2.3.1 Primeros conceptos

Primeramente vamos a definir una serie de términos que nos van a ayudar a definir lo que resta de proyecto, ya que son términos que van a aparecer muchas veces.

- Vertex (Vértice) – Punto en el espacio donde los bordes se encuentran, la esquina de una forma.
- Edge (Borde) – Conexión entre dos vértices.
- Polygon (Polígono) – Una cara de una forma 3d, conectando varios vértices a través de bordes.
- Hole - Agujero delimitado por varios vértices unidos entre bordes.
- Mesh (Malla) – Conjunto de polígonos que hacen una forma en un espacio tridimensional. También se refiere a los modelos o a los cuerpos.
- Triangles (Triángulos) – Un polígono con tres vértices exactos.
- Quad – Polígono con cuatro vértices exactos.

- N-gons – Polígono de más de cinco vértices.
- Pole – Vértice con tres vértices saliendo de él.
- N-Pole – Vértice con más de cinco vértices saliendo de él.

2.3.2 Teoría geométrica y polígonos.

El objeto básico usado en las técnicas poligonales es un vértice, un punto en las tres dimensiones del espacio. Dos vértices conectados se convierten en un borde. Tres vértices conectados entre ellos con tres edges, define un triángulo, que es el objeto más simple del espacio Euclidiano. Polígonos más complejos pueden ser creados por múltiples triángulos, o como un objeto simple con más vértices. Los polígonos de cuatro lados (quads), y los triángulos son las formas más usadas. Un grupo de polígonos conectados entre ellos a través de vértices en común se llaman elementos. Cada uno de los polígonos que forman parte de un elemento se llama cara (face).

En la geometría euclidiana, cualquier colección de tres puntos no colineales forman un plano. Por esta razón, los triángulos siempre comparten plano, cosa que no tiene porque ser cierta con otro tipo de elementos más complejos. Esto hace que en los triángulos se pueda determinar cuál es la normal de la superficie, un vector perpendicular que al plano que determina la orientación y es básico para operaciones como el raytracing para ver el plano, para saber cuál es la cara superior y cuál es la cara de atrás, con lo que se puede conseguir que una cara sea invisible.

Un grupo de polígonos conectados por vértices comunes se define como malla (mesh).

Para que se pueda renderizar correctamente, una malla no puede intersectar con ella misma, igual que la malla no contenga errores como doble vértices, bordes o caras.

2.3.3. Construcción de mallas poligonales

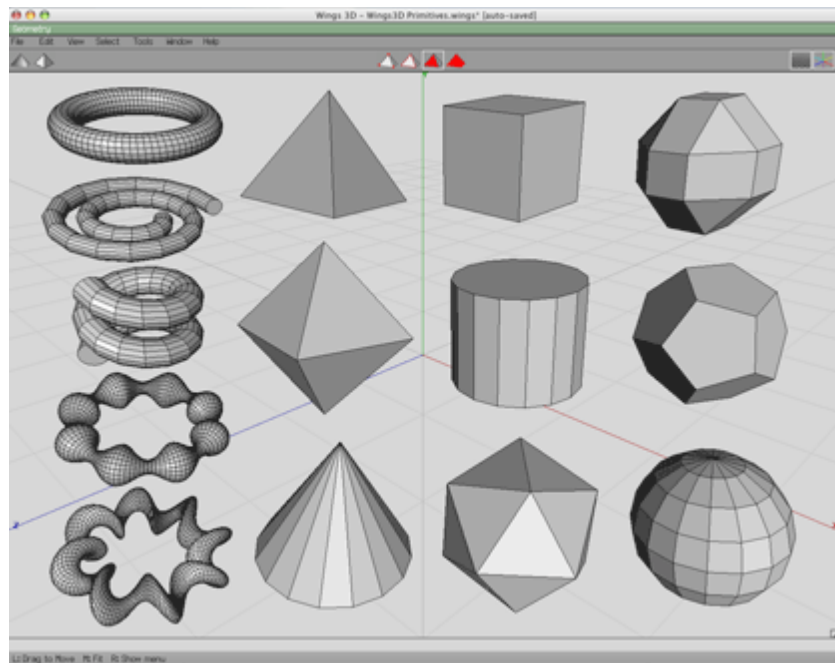
Existen varios métodos de creación de mallas poligonales, desde la creación de los vértices por separado y su posterior unión, a otras técnicas mucho más comunes usando diferentes herramientas.

2.3.3.1. Primitive Modeling

Crea modelos a partir de formas de primitivas. Es una buena forma de empezar a crear objetos simples, como casas o satélites, y luego combinarlos con herramientas como

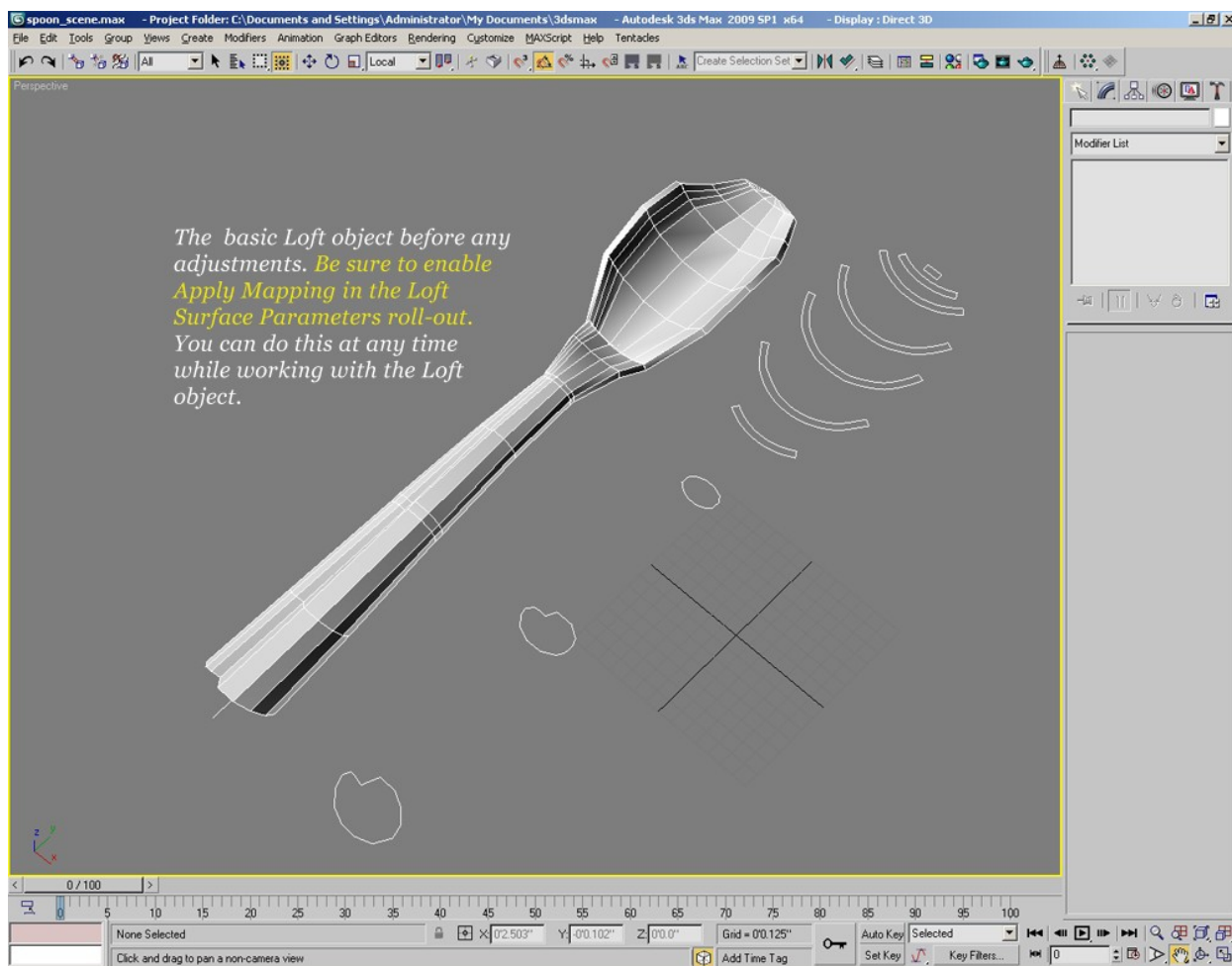
booleanos para conseguir objetos algo más complejos. También sirve como base para poder complementarla con otras técnicas bastante más complejas. Las primitivas más comunes son:

- *Cubos
- *Cilindros
- *Pirámides
- *Primitivas 2D, como cuadrados, triángulos o discos.
- *Primitivas especializadas, como la *tetera de Utah* o *Suzanne*, el mono de Blender.
- *Esferas



2.3.3.2 Spline Extrusion

Es una técnica orientada a los sistemas clásicos de modelado. Consiste en la creación de un spline y de ahí extruir para sacar un objeto o un plano modelado con el perfil creado anteriormente. Los diversos paquetes traen sus propios modificadores para poder conseguir muchos más objetos, como por ejemplo en 3dmax el uso del modificador *loft*, que permite la modificación del objeto según las secciones de los splines. Es muy usada en el diseño industrial para la creación de herramientas, secciones o diferentes elementos de objetos mayores.



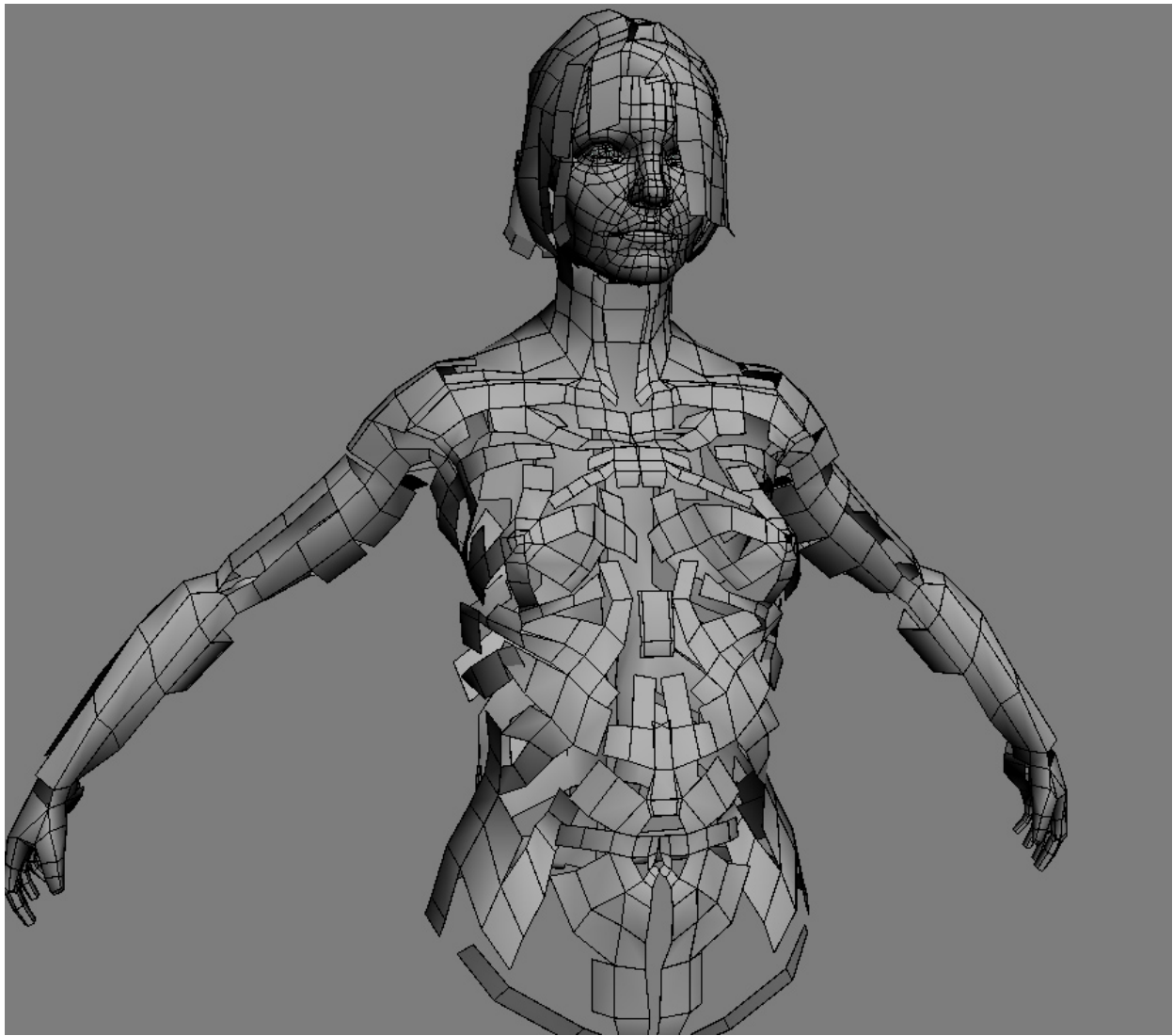
2.3.3.3 Box/Subdivision Modeling

Box modeling es una técnica poligonal en la cual el artista empieza desde una primitiva básica (normalmente un cubo) y desde el redefine su forma hasta que tiene la apariencia que desea. Normalmente se trabaja por fases, primeramente se empieza con una malla de baja resolución, se redefine la forma y luego se subdivide para suavizar los bordes abruptos y añadir detalle. El proceso de subdividir y redefinir es repetido hasta que la malla contiene suficiente detalle que se desea alcanzar. Es el método más usado de modelado, y se combina con otras técnicas, primariamente con edge modeling.



2.3.3.4 Edge/Contour Modeling

Técnica poligonal contrapuesta a box modeling. En esta técnica, en vez de empezar con una primitiva e ir redefiniendo, el modelo se construye pieza a pieza, poniendo los loops y faces a lo largo de los contornos, y luego rellenando los huecos. A pesar de que pueda parecer poco intuitivo, hay algunos objetos que es recomendable, debido a que requieren una topología y unos edge flows muy determinados, y esto te permite una precisión muy alta. La técnica consiste en modelar las partes más críticas, y luego rellenar el resto reajustando todo. En este método se suele trazar el contorno de una fotografía o dibujo, y posteriormente escoger otro ángulo y extruir esa forma 2d en una forma 3d según su contorno. Se suele usar siempre con la simetría, para tener las dos partes iguales. A pesar de ello, esta técnica en ocasiones puede llegar a ser muy liosa, como en este ejemplo:



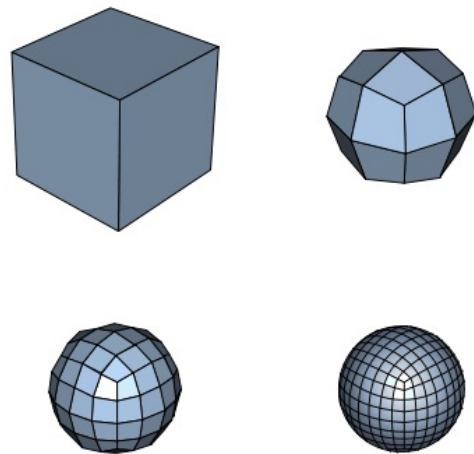
2.3.4 Subdivision Surface, explicación teórica

Ahora vamos a explicar bien los principios de esta técnica, ya que es actualmente es uno de los pilares en los que se fundamenta el modelado. Es un método de representación una superficie suave a través de la especificación de una malla poligonal mucho más basta. Esa superficie suave se calcula desde la malla como el límite de un proceso recursivo consistente en dividir cada cara en caras más pequeñas que progresivamente se aproximan más a la superficie. Comienza con una malla poligonal. Un esquema de refinamiento es aplicado a la malla, consiguiendo subdividir la malla creando nuevos vértices y caras. La posición de estos nuevos vértices en la malla son puestos según la posición de los antiguos vértices, y dependiendo cuales sean los procesos de subdivisión, estos vértices antiguos se pueden alterar basados en las posiciones de los nuevos vértices.

Todo este proceso produce una malla más densa que la original, que contiene más caras poligonales. La malla resultante se puede pasar una y otra vez por el mismo proceso anterior.

Hay varios tipos de esquemas, primero según los tipos de polígonos que usen, ya que hay esquemas orientados a triángulos y otros a quads, y dos tipos finales de esquemas, los de refinamiento y los de aproximación.

Los esquemas de aproximación consisten en que la superficie límite de las mallas se aproximan a las mallas iniciales, y los nuevos puntos de control no están en esas superficies límite. Hay varios, siendo el más conocido el Catmull-Clark, desarrollado en 1978, pero progresivamente han salido nuevos esquemas, como Doo-Sabin, Loop, Mid-Edge subdivision scheme y $\sqrt{3}$ subdivision scheme. Muchos de ellos se basan en la teoría de splines explicada anteriormente en NURBS.



Por ultimo están los esquemas de interpolación. Después de la subdivisión, los puntos de control de la malla original y los nuevos son interpolados en la superficie límite. El primero fue el Butterfly, creado por Schröder y Sweldens en 1996, pero se han ido mejorando progresivamente a otros como el midedge y el Kobbelt, ya que el primero no generaba superficies suficientemente suaves y solo actuaba en mallas triangulares, cosa que los otros ya han mejorado y se pueden hacer cargo de mallas compuestas por quads.

2.3.4.1. Ventajas y desventajas

Al final llegas a la tesitura de elegir la técnica para representar el modelo que necesitas, y tienes que elegir entre todas las anteriores. Quitando las técnicas más específicas como la procedural para paisajes, al final para hacer un modelo tienes dos representaciones posibles, NURBS y polígonos (que se pueden convertir en subdivision posteriormente).

Existen bastantes desventajas al usar polígonos para representar un objeto. Los polígonos son incapaces de representar superficies curvas, por lo que para aproximarse visualmente a una de ellas se deben usar una gran cantidad de ellos. El uso de modelos complejos tiene un coste que repercute en un descenso de la velocidad de procesado. Por

lo que en muchas ocasiones los programadores deben usar varios modelos con diferentes niveles de detalles para representar el mismo objeto y así poder reducir el número de polígonos que se tienen que renderizar.

A pesar de todo lo dicho, la mayor ventaja de los polígonos frente a cualquier otra de las tecnologías es que son mucho más veloces que el resto. Si una tarjeta gráfica es capaz de mostrar una escena con gran detalle a un frame rate de 60 fps o mayor, con NURBS y otras tecnologías similares no pueden mostrarlos a más de 10 fps.

Además existen más diferencias entre las dos tecnologías. Mientras que los polígonos son fáciles de manipular, NURBS en ocasiones son difíciles de ajustar a la forma que se quiere conseguir, sobre todo en superficies orgánicas (por ello solo se usan en diseño industrial). En las mallas poligonales a su vez se pueden sacar fácilmente los mapas UV, que son lo que se usan para texturizar, mientras que en NURBS no existe tal opción. Por lo contrario, los polígonos tienen innumerables lados mientras que NURBS solo cuatro, sea la superficie que sea. Los polígonos, como hemos visto antes, se pueden combinar con las subdivisiones, por lo que se modelan como una superficie única y se pueden programar a diferentes niveles de detalle según el nivel de subdivisión. En NURBS en cambio lo que se hace es juntar diferentes objetos NURBS (que en ocasiones son imposibles de juntar) y si se quiere cambiar el grado de suavidad, se aumenta o disminuye la teselación. Por último hay que tener en cuenta que NURBS ocupan mucho menos espacio, y son fácilmente exportables a otros programas CAT, por lo que son razones suficientes por las que aún se siguen usando en muchas áreas.

2.4 Topología

2.4.1 Tipos generales de topología

Por todo lo dicho anteriormente, nos ceñiremos a las mallas poligonales, ya que estamos buscando modelado orgánico, y en la forma más básica para animación, por lo que desecharemos el esculpido digital, a pesar de que se base en mallas poligonales también.

La próxima sección sirve para marcar las ventajas y desventajas de cada uno de los tipos de topología. La topología es la disposición de los vértices, edges y faces, así como diferentes elementos en el espacio, dando forma a la malla. A pesar de que poco a poco la tecnología sea superior, lo que suele pasar es que los tipos pasan a ser cada vez más y más estrictos. Y todo esto es porque en vez de preferir mejorar el rendimiento de las

mallas usadas en los primeros ejemplos, se prefiere mejorar que las mallas sean cada vez más y más limpias. Esto hace que la calidad sea mucho mayor. Por supuesto, hay tipos de topologías que no tienen que seguir tantas reglas como otras porque son para aspectos muy concretos y no tienen que seguir posteriores etapas de producción, como pueden ser los modelos de alta densidad (high polycount) para modelados de alta definición para muestra de detalles, que por ejemplo no necesiten animación posterior. Las iremos nombrando según el número de restricciones que imponen, de menos a más. A su vez analizaremos seis elementos que marcan mucho la topología de un objeto.



2.4.1.1 High polycount

Esta técnica está hecha para modelos que tienen un único propósito, generar texturas y generar normal maps para simular detalles. Esas texturas y



mapas son usados en otras técnicas de modelado de baja resolución, sean de animación u orientados a juegos. Otro posible uso que se le da a esta técnica es la creación de objetos que no van a ser rigged ni posed de ninguna forma, se renderizarán tal y como se han creado.

Si se usan los programas tradicionales de uso común, como Maya, 3dMax... esta es la técnica más flexible de uso, que usa reglas que en el resto de técnicas serían consideradas atrocidades. Aunque si se usan programas como Zbrush hay que tener en cuenta alguna regla más.

Nunca hay que intentar empezar a modelar con esta técnica si no se sabe cómo modelar con técnicas de lowpoly. Si el modelo no se ve bien sin subdivisión ni detalles de alta resolución, no se debe empezar a usar este tipo de workflow. Sin una base de baja resolución, un modelo de alta resolución no sirve para nada.



N-gons

Si se pueden usar, a no ser que se vaya a usar en Zbrush. En cualquier otro caso, si se pueden usar pero siempre intentando limitarlos. Si el modelo se ve bien en todos los ángulos desde los que se va a renderizar, adelante. En el caso que se vaya a exportar a Zbrush, es recomendable convertir ese n-gon a cuadriláteros y triángulos, ya que el posterior uso de la simetría fallará, ya que los pares de puntos se malinterpretarán.



Triángulos

Los triángulos funcionan bien en la gran mayoría de lugares. A pesar de ello, si se usa Zbrush hay que tener mucho cuidado. Esto se debe a que se formarán zonas tensas y deformadas alrededor de ellos, pinch-points. Estos pinch-points en la mayoría de veces no pueden ser suavizados y son causados por problemas de la topología.



Holes (agujeros)

La simulación detallada con high-polycount modeling es una técnica que permite, y en ocasiones recomienda el uso de holes. Es mucho más fácil y beneficioso usar holes cuando se está manejando un workflow (*flujo de trabajo, metodología*) que va parte por parte sin importar el resto de partes ni preocuparse del objeto como un todo. De esta forma se pueden detallar y proyectar partes de tu modelo separadas del resto, una tras otra. Además, en muchas ocasiones se prefiere tener un agujero, que rellenar el agujero (*cap*) y que los algoritmos de proyección no sepan qué hacer con ellos y acaben generando errores, haciendo que los normal maps y de texturas se conviertan en un caos.



Dense / High Polycount Áreas

En esta técnica es la única ocasión en la que se puede no prestar nada de atención al número de polígonos. El límite está dado por el hardware que se disponga, y lo bien optimizado que sea el software 3d que estemos usando. Subdividir, detallar y retener esos detalles en cada uno de los niveles de subdivisión sólo se puede hacer en Zbrush, Mudbox y en Blender si se usa la opción de Multires. El resto de software no usa el algoritmo Catmul-Clark ajustado para poder salvar cada detalle en cada nivel. El resto de algoritmos de subdivisión, como NURBS, Doo-Sabin y Lace también generan superficies suaves, pero solo se pueden usar para suavizar esa superficie, no para modificarla posteriormente.



Vértices de 5 o más lados (Npoles)

Los npoles son correctos, a no ser que se esté creando una malla de alta definición en Zbrush. En ese caso, esas áreas pueden ser malinterpretadas cuando se use un pincel 3d de gran potencia. Si esto ocurre, saldrán muchos pinch-points, por lo que quedarán zonas bastante toscas, rígidas y tensas en el modelo. Además, como hemos comentado antes, estos pinch-points no se pueden arreglar en Zbrush.



Vértices de 3 lados

Estos no dan ningún problema.



2.4.1.2 Classic (Limited) Game

El tipo de malla preferido por los motores gráficos es una malla convexa (todas las superficies salen hacia afuera, ninguna se mete hacia adentro), creada



por un número reducido de triángulos y con un simple shader como el shader suavizador (smooth shader), equivalente a los grupos de suavizado.

Los juegos 3D originales son diseñados para poder trabajar en ordenadores que no tenían tarjeta gráfica ni ningún hardware gráfico acelerador. Además, no existía ningún dispositivo que cumpliera esa función. Esto implicaba una limitación en cuantos triángulos podían ser mostrados por pantalla sin sacrificar el frame rate ni la jugabilidad.

Los quads, que son más intuitivos de manipular lógicamente, en esa época no estaban disponibles, así que para conseguir la topología cuadrilátera hay que sacrificar número de triángulos. Por lo tanto, el detalle era simulado por un texturizado muy optimizado y de aspecto muy profesional, muchas veces incluyen efectos muy simples como luces y brillos todos ellos en un mismo mapa de color. Las físicas e interactividad son colocadas antes de haber probado nada gráfico, y para optimizar las físicas, los únicos algoritmos que se usan son orientados a superficies convexas y ya están introducidos en el código. Por todo ello el modelado estaba limitado a ese tipo de curvas.



N-gons

Su uso era impensable, ya que eran capaces de bloquear totalmente el ordenador.



Quads

Su uso no estaba permitido, por lo que para poder desarrollarlos hay que convertirlos posteriormente a triángulos, así como optimizar los vértices para que funcione correctamente.



Triángulos

Es la base de esta topología, y todo estaba optimizado para su uso.



Holes

Imposible a riesgo de bloquear el ordenador.



Dense/ High Polycount Areas

No era posible, ni existía la tecnología ni los conceptos para ello.



Npoles

Eran muy recomendables ya que permitían optimizar notablemente los modelos y reducir el número de caras.

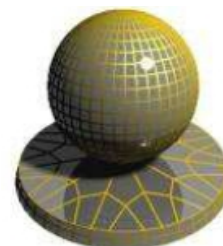


Vértices de tres lados

Su uso está permitido y en ocasiones son buenos al permitir la optimización del modelo.



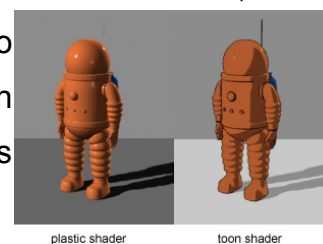
2.4.1.3 Animación



Hay muchas maneras de hacer animación, pero el flujo normal de trabajo incluye animar y hacer test de

renderización a baja resolución o en una escena simplificada, para cuando todo este correcto, subdividir toda la geometría y mejorar la iluminación, incluyendo los parámetros de iluminación indirecta hasta el límite que determine el tiempo de renderizado.

En ocasiones no hace falta subdivisión, sobre todo en casos donde hay suficiente detalle en la geometría a base de segmentación y se ha hecho manualmente, o con otras técnicas. La animación que incluye principalmente geometría de tipo mecánico no requiere gran cantidad de subdivisión. En cambio, los personajes orgánicos suelen mejorar con subdivisión, sobre todo en planos cortos. A su vez, la animación 3d de estilo cartoon, sea o no sea cell-shaded (con filtros tipo toon shaded, filtros no fotorrealistas que hacen que los gráficos por ordenador parezcan dibujados a mano), normalmente requiere subdivisión, en ocasiones más de uno o dos niveles.



La restricción para la topología de animación no suelen ser tan inclusivas como la topología de los motores de juegos modernos 3D. Hay diversas razones para ello:

1* El animador tiene control sobre todos los ángulos que el modelo va a ser observado, mientras que, en una gran cantidad de ocasiones, eso no puede ser controlado para cada ángulo que suele darse en una mecánica de juego.

2* El animador no tiene que preocuparse por las especificaciones requerimientos del motor 3d. Él puede, por ejemplo, elegir camuflar un agujero con otra geometría sin tener que preocuparse por rellenarlo.

3* El animador no está limitado a las velocidades de procesamiento 3D a tiempo real. Él puede extender el tiempo de procesamiento (el renderizado) sustancialmente, con lo que puede mejorar drásticamente la calidad de la animación.



Ngons

Los ngons generalmente tienden a estropear las animaciones y el posado de los personajes, no importa lo lejos que estén de las articulaciones. Tienden a parecer extrañas a pesar de usar algoritmos de subdivisión que “entienden” cómo procesarlos. Claramente ngons hay que desecharlos de los modelos. Hay que tener en cuenta que los modelos de alta definición no son para los renders finales ni tampoco son usados en motores de juegos.



Triángulos

En ocasiones, en animación, los triángulos son admisibles. En esos casos pueden ajustarse mejor en la geometría que otros polígonos. Un buen ejemplo de cuando es mejor su uso es cuando se necesita hacer un objeto puntiagudo, como una pica, o cualquier objeto cónico sin necesidad de una posterior subdivisión. Su uso no debe situarse en los flujos principales, ni en articulaciones o lugares donde se produce mayor frecuencia de movimiento. Por ejemplo, en el caso de la cara, no se deben poner ni en las cejas, ni en la línea de la mandíbula, ni en el puente de la nariz entre los ojos, ni en las líneas de expresión de los ojos, ni labios. Por lo general, cualquier parte que está orientada hacia otra parte.



Holes

Se deben usar agujeros en los modelos de animación. Esto se debe a que la luz indirecta, o la más avanzada, se procesa mucho mejor cuando las superficies totalmente inútiles son completamente eliminadas, con lo que se optimiza la escena. Eso sí, hay que tener en cuenta que los agujeros no deben verse en ningún momento, ya que si se ven

dan la impresión de ser una animación de baja calidad y poco profesional.



Dense / High Polycount Areas

Se recomienda usar un polycount mediano para animación, a pesar de que en determinados lugares se puede conseguir una textura más suavizada con un número mayor de subdivisiones. A pesar de ello, solo una gran compañía con grandes granjas de procesamiento se puede permitir renderizar los modelos con muchos nodos de iluminación avanzada que puedan manejar una alta resolución con buena iluminación y con unos tiempos de renderizado aceptables.



Npoles

Los n-poles son aceptables, siempre que no interfieran con los flujos principales, tal y como se ha explicado en la sección de triángulos. Se deben reducir al máximo.



Vértices de tres lados.

Estos son aceptables, como los anteriores, aunque dan menos problemas que los n-poles. No ponerlos en los flujos principales.



2.4.1.4 Juegos modernos

Esta nueva revisión sobre la topología en los juegos ha modificado una gran cantidad de cosas respecto a la



anterior. Sobre todo porque se han eliminado una gran cantidad de las restricciones que tenía la original, permitiendo al artista mayor libertad creativa, añadiendo la demanda por parte del público de mejores gráficos y mejores efectos visuales en el juego, y viendo las increíbles mejoras que ha tenido la tecnología respecto a las tarjetas gráficas, firmware y otros aspectos del pipeline del hardware, pero eso ha creado una gran cantidad de nuevas restricciones en la animación.

Ahora, los gráficos en los juegos están alcanzando a los gráficos de animación, así como los gráficos del interior de los juegos (in-game). Viendo el factor de entretenimiento, los juegos, sean de consola o de ordenador, tienen una gran ventaja respecto a una película de animación, que nunca va a cambiar. Y esto es la interactividad. Esa interactividad tiende a atraer a la gente, a convertirla en más interesante, pero todo eso lleva un coste.

Desde que los motores de juegos son interactivos, usan un motor de físicas, y este debe funcionar a la vez que el motor de renderizado cuando se está jugando al juego. Eso significa que los algoritmos de física, los algoritmos de colisión, deben funcionar en objetos que han sido creados bajo una serie de reglas.

La restricción de que las caras deben ser convexas hace tiempo que se eliminó, desde que los motores de físicas pueden elegir el algoritmo óptimo para un objeto dinámico una vez importado ese objeto. Si el objeto es convexo, se usarán los algoritmos óptimos convexos, mientras que si se usan los cóncavos, se usarán el otro tipo de algoritmos que en esta nueva generación de ordenadores ya los pueden manejar correctamente. A pesar de todo, sigue habiendo grandes problemas con los agujeros. La gran mayoría de motores no conseguirán mostrar modelos con agujeros correctamente, se mostrarán de manera extraña. Otros directamente no los aceptarán y devolverán un error al intentar importarlos. A pesar de que cubrir estos agujeros puede llevar a veces un poco de tiempo y trabajo, debido a las restricciones al número de polígonos, conviene hacerlo para tener una sola capa de superficie en la geometría. Los algoritmos de alta calidad de colisión necesitan un objeto o elemento con una superficie continua, sin roturas, porque no “saben” cómo colisionar objetos con agujeros en la geometría.

Aún hay problemas con objetos con alto número de polígonos. Esta vez, a pesar de todo, los requerimientos del número de polígonos para un motor más o menos decente se han ido doblando progresivamente desde el Quake III (este motor sigue siendo referencia en el mundo de los juegos ya que muchos se basan en este y en el del Unreal para la creación de un videojuego completo), y si llegamos a los requerimientos de los juegos más punteros esos números ya se han multiplicado mínimo por cuatro. Esto significa que se pueden poner, mínimo, cuatro veces más polígonos que antes por modelo. Unas buenas medidas son, por ejemplo: de 1000-2000 quads (x2 triángulos) para un personaje simple, de 2000-3000 para un personaje humanoide, y de 3000-10000 para un personaje avanzado, muy detallado, o un personaje muy grande que solo aparece un número reducidísimo de ocasiones (un par de ellas). El número de polígonos se debe ir calculando y monitorizando, pero siempre se debe conseguir que todo el juego funcione a un frame rate de 45-60 en cualquier punto del juego.

Esta nueva forma de topología es prácticamente aceptada por todos los profesionales, aunque hay algunos modeladores, sobre todo aquellos que modelan para animación, que enfatizan el minimizar o incluso eliminar todo el uso de los triángulos. En el otro lado de la balanza, hay profesionales de renombre, como Kevin Lanning, una de las referencias que

trabajó para proyectos como el Gears of War, que usan triángulos y consiguen modelos increíbles. Estos modelos lucen perfectamente de manera estática y dinámica. Pero hay que tener en cuenta que esos triángulos, como hemos comentado antes, se colocan en sitios donde no se van a modificar durante la animación, como ciertas partes de las botas, cascos, y en ocasiones en la parte de la piel donde la piel se pega al hueso o al cráneo. Pero estos métodos también tienen algunas pegas y deben ser manejados a la perfección.



Ngons

Como sucedía en la topología de animación, los ngons suelen estropear el aspecto del carácter, da igual lo lejos que se encuentren de una articulación.



Triángulos

Los triángulos pueden hacernos reducir el número de polígonos sustancialmente. A su vez son buenos cuando se necesitan formas punzantes en objetos rígidos. La única cosa que hay que tener en cuenta es no situarlos en líneas de expresión (grace-lines) o articulaciones.



Agujeros

Como se ha descrito anteriormente en la introducción, su uso está terminantemente prohibido. Destrozan los mapas de texturas y todo lo que se intenta mostrar si no son cubiertos. Además tampoco funcionan con los algoritmos de colisión.



Dense/ High Polycount Areas

Se recomienda usar un polycount mediano para animación, a pesar de que en determinados lugares se puede conseguir una textura más suavizada con un número mayor de subdivisiones. A pesar de ello, solo una gran compañía con grandes granjas de procesamiento se puede permitir renderizar los modelos con muchos nodos de iluminación avanzada que puedan manejar una alta resolución con buena iluminación y con unos tiempos de renderizado aceptables.



Npoles

Los npoles son aceptables en los motores gráficos actuales, siempre que no interfieran con los flujos principales.



Vértices de tres lados.

Estos son aceptables, como los anteriores, aunque dan menos problemas que los n-poles. No ponerlos en los flujos principales.



2.4.1.5 Strict Quad



El aumento de las tendencias o requerimientos de este tipo de topología ha surgido del aumento de las ventajas que la topología de quads transmite. Si se modela sin comenzar desde ningún triángulo o ngon, se sigue un flujo de trabajo (pipeline) estándar (comenzar desde un modelo low-res, pasar a uno high-res, de ahí introducir texturas, huesos..., de ahí animar, y por último, opcionalmente, exportarlo a un motor de juego) y no se hace ningún error estúpido, entonces no hará falta debuggearlo para ver si funciona todo bien a la perfección.

Con el aumento de las capacidades de los polígonos en el renderizado y en los motores de juego, la necesidad de optimizar los modelos al mínimo absoluto de triángulos estaba ahí, y se desarrolló poco a poco.

Siguiendo con el ejemplo de la sección anterior, Kevin Lanning y una parte de sus compañeros han demostrado que el uso de algunos triángulos en los modelos de juego no está aún desfasado, pero hay que recordar que el pipeline de Kevin no es lineal como el que estamos mostrando antes. De hecho, se podría considerar lo contrario a lo normal. Kevin usa a la vez 3dMax y Zbrush para conseguir un nivel de detalles enorme en modelos de muy alta densidad. Aunque hay que tener en cuenta también, que Kevin siempre intenta trabajar con modelos con topología de quads perfecta. Como suele ocurrir, siempre se intenta reusar, con los debidos reajustes, partes de modelos ya usados. De ahí, con las herramientas proporcionadas por 3d Max, pasa de un modelo low-res a uno de alta resolución. Posteriormente usa una herramienta, llamada Polygon Cruncher, que optimiza el modelo proporcionado por Zbrush y reduce el número de polígonos a un número que 3d Max pueda manejar correctamente, sin perder un ápice de calidad. Después se reajusta manualmente para minimizar las diferencias entre los dos modelos.

No le importa introducir algún triángulo, siempre que se sigan las líneas marcadas dichas en secciones anteriores, como no modificar las líneas de animación. De ahí define los mapas UV en la mitad y una tira de polígonos siguiente. Borra el resto, que será un poco

menos de la mitad. De esta manera se previene que haya errores en el área de la unión de las dos mitades. Posteriormente se realiza la extracción de los mapas de normales, otras texturas y shaders necesarios del modelo de alta resolución al de baja. Esto proporciona suficientes mapas para poder combinarlos entre ellos y alcanzar un gran nivel de detalle al completar el modelo con la simetría.

A pesar de todo lo dicho, el flujo de trabajo normal suele empezar creando el modelo en baja resolución, de ahí sacar los UV, exportarlo desde Max para pasarlo a Zbrush, detallar todo el modelo y unirlo todo. Este flujo de trabajo tiene una serie de ventajas:

1º Si unes todos los mapas (proceso llamado *bake*) en Zbrush, no hay que preocuparse en reajustar el modelo de alta resolución en el de baja. El truco es que el de baja se reajusta automáticamente cuando se está trabajando sobre capas más altas, así que directamente la versión de baja resolución es una forma optimizada de la de alta resolución. Si la de alta resolución no se ajusta en el modelo de baja que fue escogido en 3dMax o Maya, y se prefiere ese específicamente, hay varias formas de importar ese modelo a Zbrush sin perder ningún detalle de capas superiores.

2º Si se usan las herramientas específicas de Zbrush para sacar los normal maps, o se usa Zmapper, no hace falta que se espere un gran tiempo para que te los saque, ya que el código está muy optimizado.

El mayor problema es que no se puede tener ningún triángulo ni ngon cuando se importa a Zbrush, o lo más normal es que se estropee el modelo y no lo consigan cargar.



Ngons

Está prohibido por varias razones, incluyendo la incompatibilidad con la subdivisión Catmul-Clark usada en Zbrush y los no queridos estiramientos o apelotonamientos en animación.



Triángulos

No. La topología se llama Strict Quad por algo, y es por no admitir ningún otro tipo de polígonos en ninguna fase del proyecto.



Agujeros

No, por la misma razón que la anterior sección.



Dense/ High Polycount Areas

No, debe seguir las mismas pautas que la topología de juegos modernos, en la que no debe haber ni un solo polígono creado sin un propósito específico. Tampoco nos podemos guiar por el número de polígonos, ya que lo que se considera muy denso para juegos, es posible que no sea nada denso para animación.



Npoles

Estos vértices siguen siendo válidos, siendo este y los vértices de tres lados los únicos elementos que se deben tener en cuenta en los flujos de expresión. Por supuesto su existencia es peligrosa y por eso hay que limitarlos lo máximo posible.



Vértices de 3 lados.

También son válidos, pero con las mismas precauciones que en punto anterior.



2.4.1.6 Box Topology



Esta es la topología más especializada de todas, y es necesaria tener mucha maestría para poder aplicarla correctamente. En ella se prohíbe el uso de todos los

elementos que pueden causar problemas más tarde, sin contar, por supuesto, con malas proporciones, formas incorrectas o diseño incorrecto. Esto es algo que al final todos los modeladores acaban llegando en algún punto de su carrera.

El flujo de trabajo suele empezar comenzando desde una caja de un solo segmento o desde otro modelo que haya seguido esta topología anteriormente, como un humano genérico, o cualquier otro tipo de modelo. Este modelo debe de tener ya las coordenadas de texturas (Uvs) ya definidas, ya que es más fácil ajustar Uvs que crearlos desde cero.

En el primer caso, uno se intenta restringir a segmentar la caja a lo largo y ancho de todas las caras, y en muchas ocasiones se usa simetría para ayudar en este proceso. Los segmentos suelen ser llamados *edge rings*, mientras que el concepto segmentos se refiere a la tira que se queda entre dos *edge rings*.

En el último caso, se va ajustando el modelo asignado, añadiendo más geometría en los lugares que más se necesitan con herramientas como *edge ring tool* en Maya o la correlativa en 3dMax, *edge connect tool*. Usando este tipo de herramientas no se suele

tener problemas posteriores con los mapas de coordenadas existentes, como máximo se tienen que relajar un poco después de editarlos.

Si se empieza desde una caja, se empieza sólo con ocho vértices de tres lados, por lo que no hay agujeros, ni triángulos, ni ngons, ni áreas densas, ni npoles, ni ningún tipo de objeto no deseado. Empiezas desde un estado limpio, y no importa lo que hagas, si se sigue la metodología al pie de la letra, no se tendrá ninguno de esos problemas. Además, el proceso para definir los Uvs será mucho más fácil. No hará falta usar técnicas como pelt-mapping en la gran mayoría de casos, con usar lo más simple, en unos cuantos minutos estará hecho. Incluso con un modelo creado anteriormente con Uvs, este proceso sólo durará unos cuantos segundos.



Ngons

Completamente prohibidos.



Triángulos

Completamente prohibidos.



Agujeros

Completamente prohibidos.



Dense/ High Polycount Areas

Cada polígono debe contar, cada polígono debe tener su propósito, incluso cada vértice lo debe tener. Si no lo tiene, todo ese flujo igual no debería estar situado ahí.



Npoles

No deberían darse en el modelo. Si se da, esos lugares son difíciles para definir posteriormente los Uvs. Además de los problemas ya mencionados en los modelos anteriores.



Vértices de 3 lados.

Estos deben mantenerse al mínimo. Cuando se empieza desde una caja, se empiezan con ocho de estos, las esquinas de la misma caja, por lo que no hace falta agregar más.

2.4.2 Requisitos mínimos

En esta siguiente sección vamos a analizar los requisitos mínimos que se consideran en un modelo para decir que un modelo es de calidad, sin tener en cuenta que luego el modelo sea bonito o no, es decir, lo artístico. Para ello vamos a plasmar lo que exige Turbosquid, el mayor mercado de compra/venta de modelos 3d online (mostramos todos los requisitos, no solo los de modelado, ya que en la siguiente sección también se engloban algunos de estos requisitos):

1. Geometría

- 1.1 No puede contener vértices aislados. También se incluyen vértices que solo tengan dos lados y no sean esquinas, que hayan sido borrados los lados pero no se hayan borrado los vértices creados.
- 1.2 No puede contener vértices coincidentes.
- 1.3 No puede contener caras coincidentes o que se superpongan.
- 1.4 Las normales de las caras deben señalar hacia afuera, y para ello nos referimos a que no estén mal señaladas ya que el renderizado no lo realizaría correctamente.
- 1.5 No se deben tener objetos vacíos. Todos los objetos de la escena deben tener geometría o deben ser helpers de algún tipo.
- 1.6 La geometría no debe tener bordes abiertos que causen poder ver a través del modelo cuando se renderiza.

2. Topología

- 2.1 Solo se pueden entregar modelos que estén compuestos por quads y triángulos.
- 2.2 Los modelos deben usar en la medida de lo posible los quads. El máximo permitido estará en un rango entre el 10% y el 20%, aunque hay algún modelo en concreto que puede subir un poco más dependiendo de la forma de la topología. Hay que tener en cuenta que los modelos hechos para realtime tienen que tener un ratio de triángulos mucho mayor para conseguir los requisitos de la mínima geometría posible para hacer el proyecto.
- 2.3 Deben de tener un flujo de lados limpio. Los modelos deben tener el mejor edge flow, con la suficiente geometría para poder llegar a los requisitos sin necesidad de geometría innecesaria.
 - 2.3.1. Topología subdividible: El modelo puede ser subdividido limpiamente para alcanzar mayor detalle y aún deberá mantener la forma.

2.3.2. Los vértices no deben tener más de cinco lados en superficies curvadas.

2.3.3. Los lados de apoyo deben servir para mantener la forma después de la subdivisión.

2.3.4. Excepciones:

2.3.4.1. Existen excepciones para objetos pequeños o insignificantes de la escena. Esos objetos no entran en el criterio dicho anteriormente. Ejemplos de esos objetos pueden ser tornillos, tuercas, bobinas, alambres... que son pequeños en comparación con el resto del modelo, y no hacen falta que cumplan las reglas de topología limpia.

2.3.4.2. Existen excepciones para objetos de un lado con mapa de opacidad. Estos objetos que representan objetos finos de la vida real, pero representados a través del mapa de opacidad no hace falta que cumplan las reglas anteriores. Ejemplos de estos pueden ser plumas, pétalos, papel o tela. Esta excepción no se aplica a objetos tridimensionales tan

2.3.4.3. Existen excepciones para los modelos a tiempo real: Los modelos dirigidos a motores de juego, ha visualizado a tiempo real o elementos de fondo que deben tener el mínimo número de polígonos pero deben mantener en la medida de lo posible la forma. Para estos modelos, está permitido un porcentaje alto de triángulos, mientras eso consiga reducir al máximo el número de polígonos.

2.3.4.4. El valor del crease en 3dMax y Maya debe ser cero para todos los lados. Este requerimiento asegura que un modelo, al ser convertido a otros formatos, mantendrá la misma forma cuando sea subdividido. Esto también se aplica a los modelos de tiempo real porque no hay razón para que este valor sea diferente.

3. Escala del mundo real

3.1 La escala de los objetos debe variar de 1%-3% de la de su homólogo en el mundo real. Los modelos pueden usar cualquier tipo de unidades para conseguir los valores del mundo real. Si el modelo no necesita una medida exacta (como un modelo humano o un coche sin marca), el modelo deberá tener unas medidas similares a otros objetos de su misma gama.

3.2 La excepción se dará en modelos de tamaños extremos, sean muy grandes o

muy pequeños. Los modelos de nivel microscópico o astronómico, como amebas o sistemas solares están exentos de cumplir la escala natural.

4. Posición y centrado

4.1 La base del modelo debe estar en el 0,0,0

4.2 El modelo debe estar apoyado sobre el plano de tierra.

4.3 Debe estar orientado hacia el arriba global.

5. Transformaciones

5.1 Los ejes de transformaciones y rotaciones deben estar bloqueados

5.2 La escala debe estar al 100%

6. Nombrado y organización

6.1 Deben de tener nombres únicos y descriptivos, no valen los nombres por defecto. Para huesos, se deberá incorporar un prefijo delante de cada uno.

6.2 Debe incluir una jerarquía o un agrupado para todos los objetos del modelo.

6.3 Deben tener un nombre corto y fácilmente escaneable.

2.4.3. ¿Por qué es importante la topología?

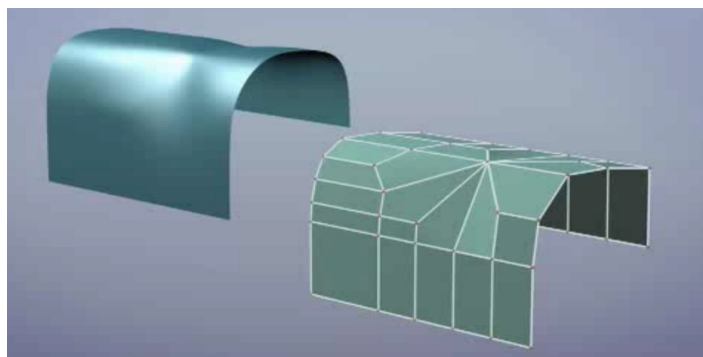
Posiblemente sea el aspecto individual más importante en la creación de cualquier modelo. Al fin y al cabo es la estructura de la malla y el modo en que esta fluye a través de la superficie y detalles del modelo. Al ser una materia tan amplia, hay muchos conceptos que hacen falta para poder entender cómo crear una malla limpia y una topología eficaz para cada uno de los modelos, del tipo que sean.

Hay una serie de razones por las que es básico su uso.

1ª Para tener áreas de deformación limpias. Al animar caracteres y otros objetos orgánicos es muy importante obtener deformaciones y movimientos limpios. Si eso no se consigue, la atención al visualizarlo se centrará en ese defecto y distraerá del resto de la animación. Si se estructura de una manera efectiva, seremos capaces de obtener mejores deformaciones con mucho menos trabajo. La idea al trabajar con modelos de personajes, es distribuir los edgeloops de manera que sigue el contorno de los músculos y de otras formas maestras. Luego se desarrollará este concepto en profundidad.

2ª Para mejorar la calidad de la superficie de un modelo. Es un aspecto crucial, ya que todos los aspectos deben estar debidamente controlados. Todos los modelos deben estar suavizados, desde los que son curvas a los bordes más afilados. Esta diferencia puede causar que un modelo pase de un modelo decente a un gran modelo.

La clave está en dos conceptos, remarcando los detalles claves con una serie de edgeloops que permiten delimitar cada parte mejor, o el uso de vértices de cuatro lados en las partes pequeñas y transiciones. De esto último sacamos lo que en ocasiones pasa cuando no seguimos estas indicaciones, que es la creación de “artefectos” que estropean la malla.



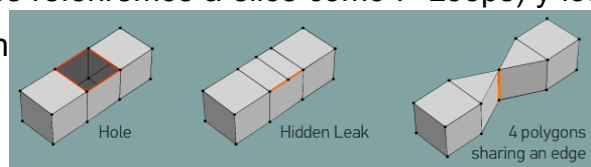
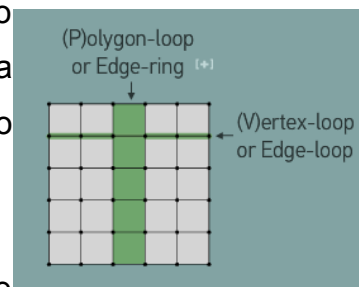
3º Podemos seleccionar y manejar modelos con mucha mayor facilidad. Esto se debe a que los programas 3d están diseñados para poder interactuar con diferentes elementos, como los edgeloops, que hacen que puedas controlar más fácilmente las proporciones. Además de poder intercambiar partes similares y poder reutilizar los modelos que has realizado previamente.

2.4.4 Conceptos básicos

La topología, como hemos mostrado antes, es el estudio de las mallas, de las conexiones entre los diferentes vértices, lados, flujos y demás componentes que determinan las mallas. Absolutamente todos los modelos 3D se basan en los conceptos de la topología, por lo que es una herramienta básica. Además, la topología se usa en 3d en flujos de trabajo, árboles de nodos, jerarquías de Rig... Y tiene mucho que ver con procesos tan importantes como Retopología (Reordenamiento de los vértices y su optimización), Stepping o subdivision, que ya la hemos definido antes.

Para ello vamos a ir desgranando uno a uno los conceptos.

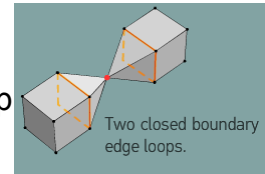
En las mallas se dan dos tipos de entidades principales, que son los Polygon-loop (o Edge-ring, o también nos referiremos a ellos como P-Loops) y los Edge-loops (o los Vertex-loop, o V-Loops). Con ellos se manejan todas las formas de una malla.



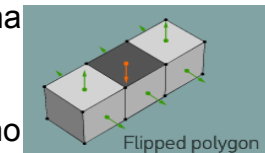
Ahora el concepto de una malla de superficie de tipo 2-Manifold. Para que esto se dé se tienen que dar los siguientes conceptos:

* Los edges deben ser siempre compartidos por dos polígonos. Si esto no sucede, se formarán agujeros o superficies ocultas.

- Los edges adyacentes a un vértice deben formar un sólo loop cerrado.

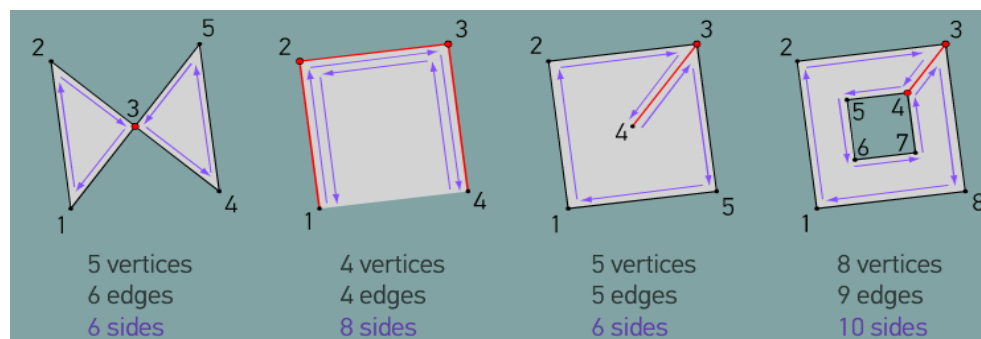
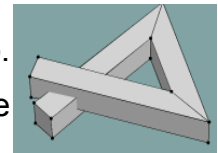


- Todos los polígonos deben estar orientados hacia la misma dirección que sus adyacentes.



- No debe existir la auto-intersección, ya que si se da, la malla no coincidirá con el volumen.

El siguiente concepto es el uso de un vértice y un edge por polígono. Cuando los vértices y los edges son usados más de una vez, el número de lados de un polígono no coincide con el número de vértices o bordes.

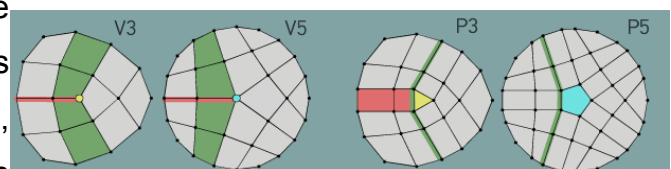


Uniando estos dos conceptos, llegamos al concepto de valencia. La valencia de un vértice es el número de edges que inciden en él, mientras la valencia de un polígono es el número de edges que lo componen.

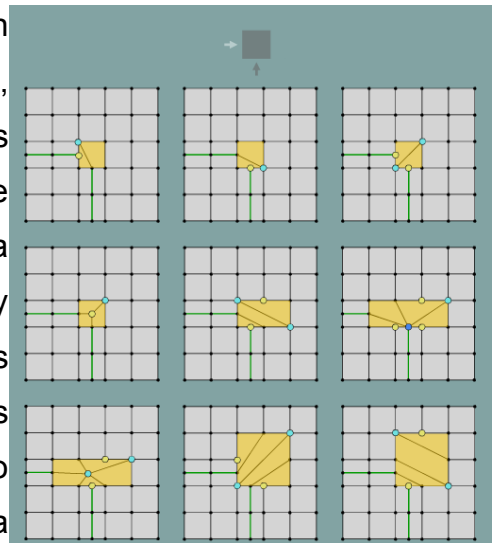
Valence of Vertices and Polygons						
(n)	2	3	4	5	6	>6
(V)ertex	●	●	●	●	●	●
(P)olygon	—	▲	■	◆	◆	◆

Normalmente, como hemos visto en los tipos de topología anteriormente, suele usar mallas regulares compuestas sólo por V4 y P4, debido a las ventajas que suponen al ser tan predecibles en aspectos como la subdivisión, el mapeo o las deformaciones.

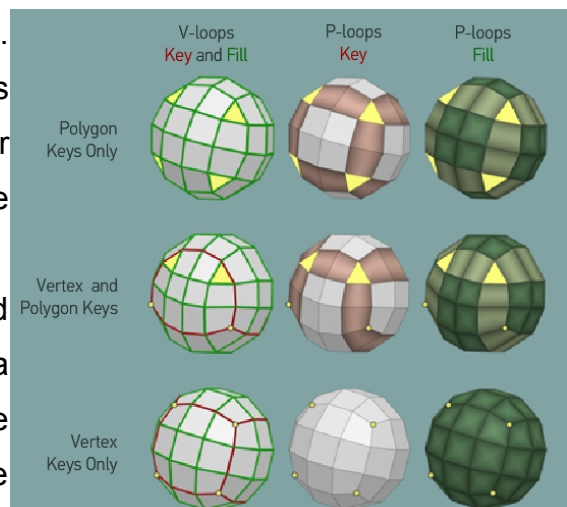
Aquellos vértices y polígonos con valencia no igual a 4 son anomalías o Keys (o Poles), y tienen la particularidad de parar los loops de valencia 4. Por ejemplo, los Key vértices paran los V-loops (los loops de vértices), mientras que los polígonos Keys detienen los P-Loops.



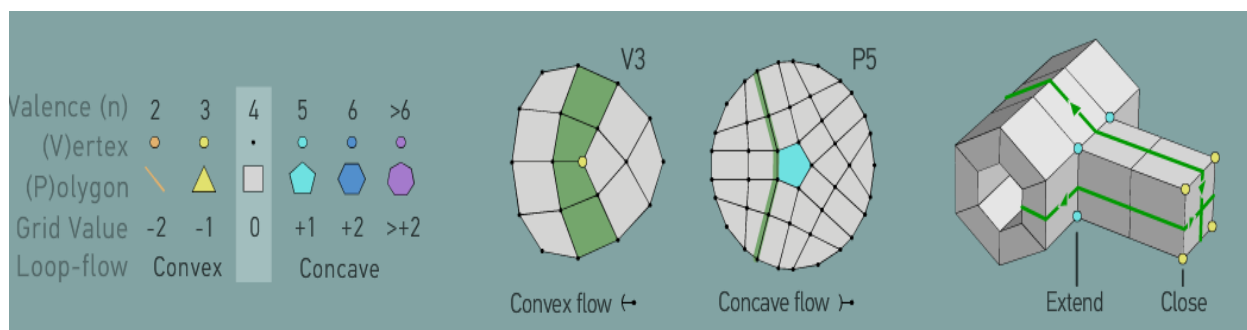
Esos loops interrumpidos son los Key-Loops, que son los que definen la topología estructural de una malla, mientras que los cerrados son los Fill-Loops (completos), que son simplemente los que se encuentran entre los keys. Si todos los Keys de una malla son vértices, entonces los polígonos serán P4 y todos los P-Loops serán Fill (todos son quads). Si todos los Keys de una malla son polígonos, todos los vértices serán V4, por lo que todos los V-Loops serán fill. Como hemos visto en las prácticas de modelado, se busca tener todos los polígonos quads, además de las razones anteriores, por otras dos. La primera es que las herramientas están orientadas siempre hacia los loops de polígonos, por lo que se pueden manejar más fácilmente. Además es que los vértices Key son mucho más fáciles de controlar ya que están compuestos por un sólo elemento, no como los polígonos, que están compuesto por varios.



El siguiente concepto es el de Grid Value. El Grid Value (GV) de un elemento es igual a su valencia menos 4, ya que en una malla regular en la que todos sean P4 y V4 la valencia total es 4, y eso se toma como referencia para el GV y se le da valor



0. En el caso de uno de 3, el GV será -1, y en cambio uno de 5 tendrá GV+1. Si nos fijamos en la malla, aquellos elementos que provocan que el GV sea positivo hacen que la malla sea cóncava, mientras que los que son negativos hacen de la malla convexa.

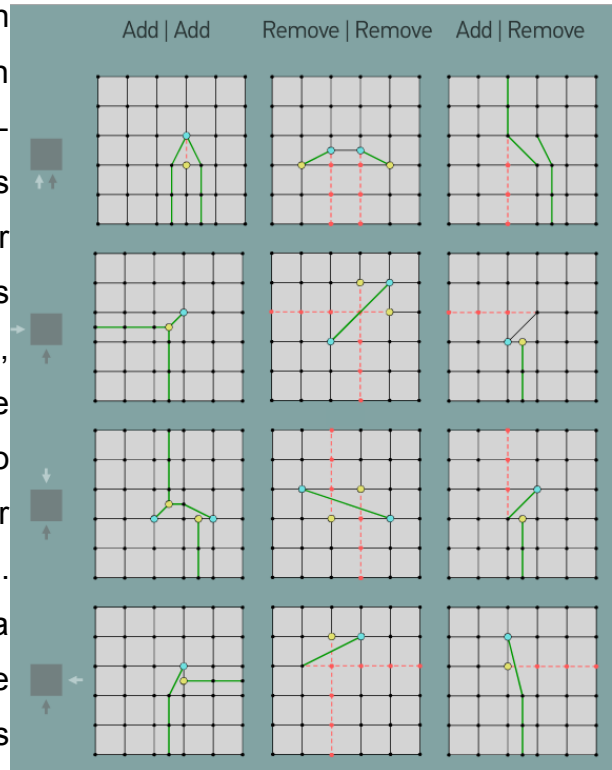


Todo esto tiene una propiedad, que es que el GV siempre es constante, si el número de vértices es constante, eso hace que si los bordes de una malla tienen un GV determinado,

puedes calcular las posibles combinaciones de elementos que se pueden poner en el interior ya que la suma de todos los diferentes GV locales será la del GV original.

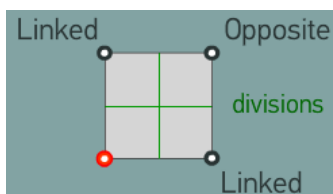
Una vez que tenemos definido el concepto de Key, ahora hay que saber usarlo a nuestro favor.

La operación principal que se puede hacer con estas Keys es poder transferirlo (shift), poder darles las diferentes propiedades a un elemento vecino. Para poder hacerlo se usan los principios de editado de parejas de edge-loops. El editado de parejas de edge-loops es añadir y/o quitar dos edge-loops para alcanzar una región determinada. Esto crea las condiciones para cambiar el flujo de loops, trasladar Keys y poder controlar la densidad de la malla en determinadas áreas. Un aspecto problemático es que los loops pueden acabar propagados fuera de la zona que nos interesa. Al hacerlo en parejas en ocasiones genera loops extra, pero nos aseguramos que se cumpla de que siempre sean quads. En los ambientes que solo soportan quads y triángulos, las líneas verdes serán las primeras en ejecutarse, y posteriormente se procederá con las rojas.

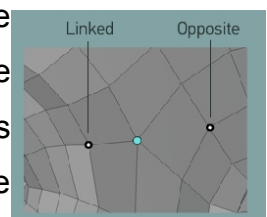


Esos loops crean las condiciones, pero el núcleo de cómo se intersectan es lo que define el flujo resultante. Una misma acción puede crear diferentes flujos.

A pesar de todo esto, como hemos dicho anteriormente, los edge-loops pueden propagarse más allá de la zona en la que estamos operando, y el trasladar las keys por parejas puede hacer disminuir este efecto, ya que los loops creados entre dos Keys se cierran.



Antes de explicar en profundidad este concepto, vamos a explicar el concepto de linked y opposite respecto a un vértice. Estos términos sirven para describir la relación entre



vértices o un Key Shift. Linked es un vértice próximo al que esté conectado con un edge, mientras de opposite no lo está, ya que se encuentra en la otra

esquina del polígono.

Dicho esto, seguimos con el concepto de mover por parejas. Cuando se hacen a través de un eje, los nuevos edge loops, se hayan añadido o retirado, se paran entre ellos, lo que hace que el cambio sea local y no se propague al resto de la malla. Trasladar por parejas con la misma valencia hace que tenga efecto espejo, mientras que parejas de un Grid Value inverso (V3 y V5 por ejemplo) hacen que se propaguen todos hacia la misma dirección.

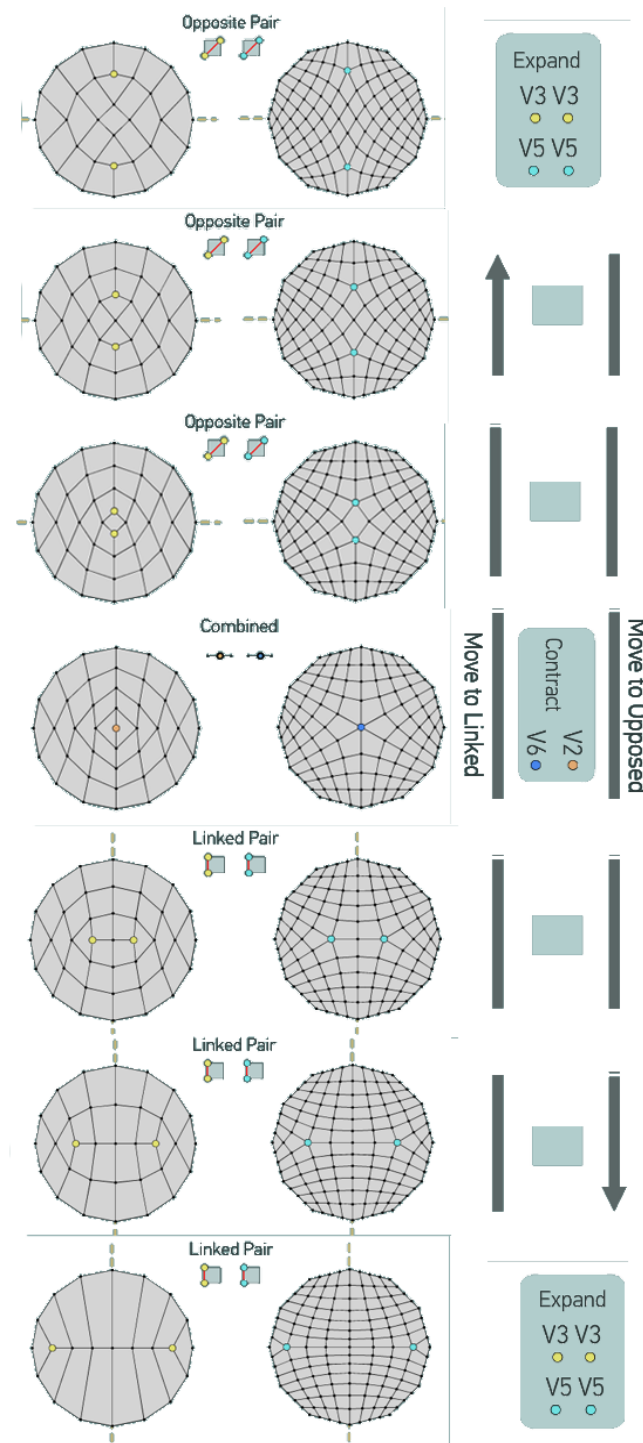


Tabla para mover diferentes vértices

	Movement Position	Add Two Edges 	Remove Two Edges 	Add and Remove Edges (mixture) 	Add and Remove Edges (alternative)
Linked					
3 Pole					
Opposed					
Linked					
5 Pole					
Opposed					

	Up	Down	Sides (Spin Edge) 	Diagonals Up 	Diagonals Down
Linked					
3-5 Pair					
Opposed					

2.4.5 Stepping

Stepping, sea hacia arriba o hacia abajo, es cambiar el flujo de loops para poder controlar la densidad local de una malla. Se genera donde se conectan zonas con diferente número de bordes, como zonas creadas con operaciones booleanas, o creadas con operaciones de Patch/Bridge, que limita la propagación de los llamados fill loops y el nivel de detalles y de limpieza.

Por ejemplo, si un área está preparada para soportar un mayor nivel de detalles, su densidad es aumentada con una mayor cantidad de edge loops. Por el contrario, si el detalle creado al principio y el número de loops deben ser limitados, la densidad se reduce. Puede producir el mismo resultado, pero usando diferentes perspectivas.

Hay muchas posibilidades, y suelen darse en la situación común de que dos bordes de edges, de diferentes densidades, son unidos entre sí.

La estructura de stepping se divide en dos variables:

*Stepping Factor= M/L

*Stepping Differential= $M-L$

Siendo M el número de vértices que contiene la malla densa, y L el número de la malla más ligera. Hay tres posibles situaciones:

1º En el caso que se añada un loop link, el diferencial se mantendrá y se disminuirá el factor



















2º Si se añade un Closed loop, el diferencial aumenta, y aumenta también el factor.

3º Si se subdivide, se aumenta el diferencial y se mantiene el factor.

Boundary's Vertex Number	4	5	6	8	10	12	e (Extrapolation)
Odd Step Amounts		$\frac{2}{1}$ $\frac{2}{1}$	Odd Step Amounts result in a triangle inside the margins. This is something to take in account in the whole process of stepping, since a bad connection or division may lead to this.				$\frac{e}{e}$ $\frac{e}{1}$
FourPointTriangle Adding Link Loops only for Constant Step Amount of 2	$\frac{2}{0}$	Best method for straight flat steps. All other methods create vertices between margins.				$\frac{e}{e-2}$	No vertices causing pinching between margins. Tough, there will be pinching on the margin. If it troubles, add an Edge Loop between the margin and the stepping to hold the mesh.
Trapezium (works in pairs) Adding Link Loops only for Constant Step Amount of 2	$\frac{1}{0}$	Stable flow, with vertices between margins to shape the mesh.				$\frac{e}{e-2}$	The placing of Link Edge Loops, in or out of the pair of Trapeziums, control the amount of vertices between margins.
Stepping Structure Stepping Elements More Edges Margin (M) $M - L = \text{Step Amount}$ Less Edges Margin (L) $\frac{M}{L} = \text{Step Factor}$ Edge Loop Types Close Loop (blue) Link Loop (green)		For modeling between margins there are many solutions, depending of the desired flow. Here are some examples but the possibilities are infinite.					$\frac{e}{e}$ $\frac{e}{e}$ $\frac{e}{2}$
Basic Flow. Plus Edge Loops to control Stepping's Amount and Factor. $\frac{3}{1}$ $\frac{4}{2}$ $\frac{5}{1}$ $\frac{e}{e-2}$ $\frac{e}{1}$		Random Example of Combining Steps $\frac{14}{6} = \frac{4}{2} + \frac{3}{1} + \frac{1}{1} + \frac{6}{2}$ $\frac{14}{6}$					
		Many times it's easier and better to divide the stepping operation in small ones. So there is more control of each area's shape and the math gets easier. Divisions might create two Odd Step Amounts from one Even Step Amount, so they have to be well placed.					

Polygon Table for 3D Modeling

Different mesh configurations within the same boundary

Boundary's Vertex Number	4	5	6	8	10	12
Central Division						
Parallel Division						
Grid Division						
<div> <div>■ Quads</div> <div>■ Squares</div> <div>■ Triangles</div> <div>■ Standing on Element</div> <div>■ No. Vertices</div> <div>■ No. Polygons</div> </div>	<div>Central</div> <div>Two Quads sharing two edges. Two FourpointTris.</div>	<div>Odd Boundary's Vertex numbers result in a triangle.</div>	<div>Central</div> <div>The 3Edge Pole.</div>	<div>Central</div> <div>Mirrored Flow. Lowest number for round meshes.</div>	<div>Central</div> <div>The 5Edge Pole.</div>	<div>Central</div> <div>Enclosed Square with mirrored flow.</div>

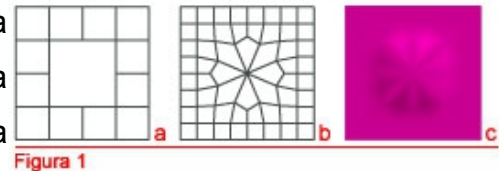
2.4.6 Edgeloops

Todos estos conceptos básicos nos permiten realizar cualquier tipo de operación en la malla. Ahora solo hace falta configurarla, darle forma, que sea funcional y usar el mínimo número de polígonos posibles para conseguir lo que necesitamos. Para ello no hace falta sólo con saber diseñar y que el modelo tenga muchos detalles. Para modelar bien hace falta un que sea un modelo funcional, y para eso se tienen que usar los edgeloops. Tiene sus contras, ya que el proceso de modelado se duplica o triplica en el tiempo. Pero consiguen su finalidad.

La técnica de los edgeloops ha ido evolucionando a la vez que han evolucionado los ordenadores. Al principio los modelos tenían movimientos robotizados, así que hacía falta mejorar ese aspecto, hacer que las mallas se deformen en consonancia con los movimientos que se creen, sin deformarlos y que queden evidenciados. Y para ello se debe recrear el movimiento de los propios músculos. Por eso esta técnica es más usada en modelado orgánico que en modelado inorgánico. Resumiendo, es una técnica de

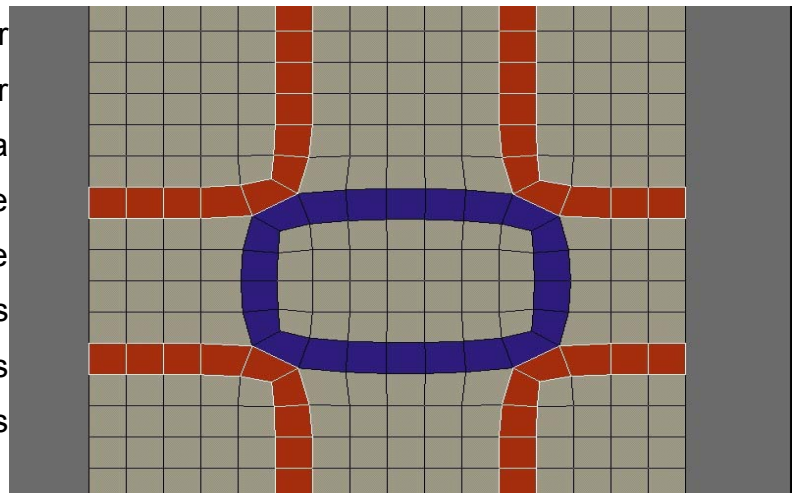
modelado cuya máxima premisa es que las aristas de un modelo deben seguir en forma cíclica, como las fibras de los músculos.

Al mover nuestros cuerpos, decimos que relajamos o tensionamos los músculos. Esta idea de relajar o tensar está presente en las dos etapas del modelo 3d cuyo edgeloop nos ayuda perfectamente. La primera etapa es la de la tensión poligonal que se debe controlar en la construcción del personaje. La segunda es la tensión del movimiento propiamente tal, donde los edgeloops ayudan a la correcta deformación de los polígonos. La tensión poligonal es lo que hemos estado analizando hasta ahora, con los poles y los ngons, que acaba desembocando en la creación de artefactos en la malla, como se puede observar en esta imagen:



Según la segunda premisa, los edgeloops son un auxilio al movimiento y deformación de una malla, por lo que deben ser usados donde el modelado del personaje se puede deformar. En las áreas donde no se deforma no hace falta, y hasta se pueden tener polígonos de 3 y 5 lados, o npoles, sin dañar mucho al personaje, eso sí, siempre sabiendo lo que se hace y reduciéndolos al mínimo.

Dónde situar los edgeloops es el trabajo del artista, y depende de su creatividad y de su pericia para concebir nuevos loops basados en los posibles movimientos del personaje creado. Y para ver si funciona ese edgeloop creado, hacen falta pruebas y más pruebas. La primera etapa del modelado es mental, siempre antes de modelar nada, es necesario conocerlo. Lo que hay que hacer es diseñar el personaje, marcar encima de él la ubicación tentativa de los diferentes edge loops que se necesitan. A su vez también se puede uno imaginar los movimientos y deformaciones posibles para modelarlos mentalmente.



Los diferentes edgeloops se forman con la disposición de poles de diferente valor en la malla. Esto hace que se creen las diferentes formas. Existen varias formas de crearlos, desde extruir vértices, a manualmente usar herramientas como cut, o herramientas como Inset o chamfer.

Tenemos loops simples, creados por estas dos últimas herramientas nombradas (sería el azul en la imagen anterior). Hay que tener en cuenta que si creamos un determinado loop, se suelen crear loops alternativos al que se está creando, o por lo menos modificando los ya existentes. También tenemos loops en U, que son perfectos para lugares como las puntas de los dedos, o simplemente para unir partes del modelado con diferente número de vértices entre sí y así reducir progresivamente su número.

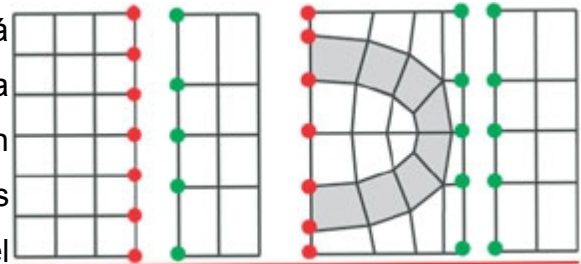
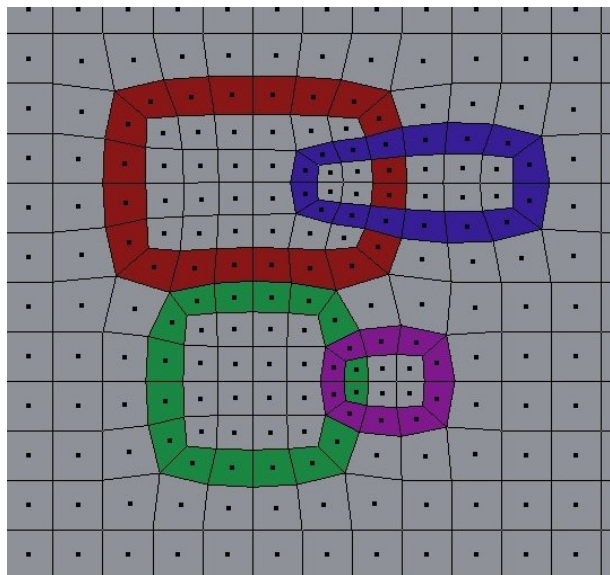
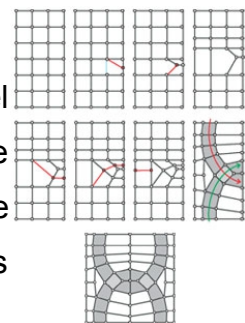


Figure 10

Los loops se pueden cruzar entre ellos, no son delimitaciones, por lo que se pueden combinar entre sí para crear todas las formas que necesitemos.

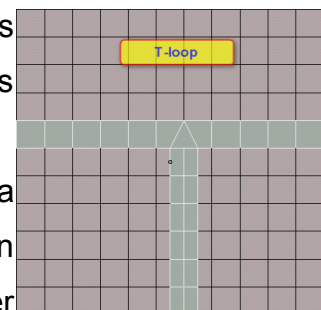


Al existir esta posibilidad, existe otro tipo general de loops que es el loop en X. Este loop se caracteriza por cruzarse entre sí, y es uno de los más complicados. Es usado en codos y rodillas, y a pesar de que en plano parece fácil, una vez que se hace en volumen se pierden las referencias y puede confundir al modelador.



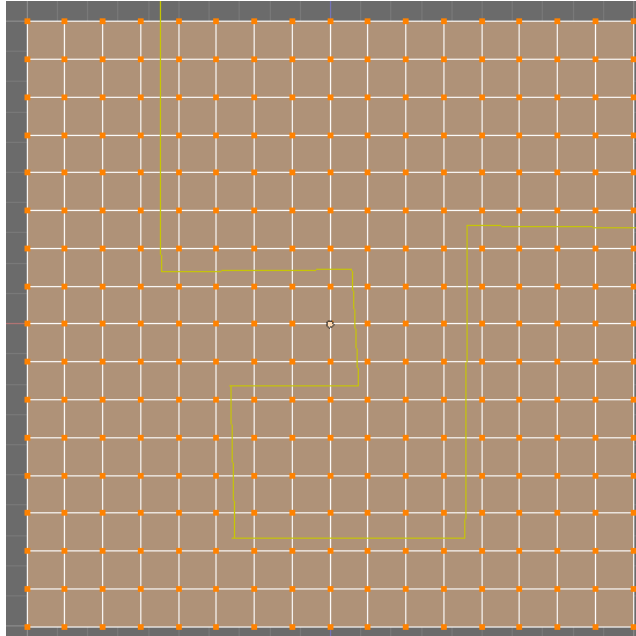
El último tipo remarcable de loops que se pueden obtener son los T-loops. Son loops en forma de flecha creados al unir los V3 y los V5. Son usados para añadir detalle, como puede ser en el cuello.

Por último cabe destacar es el uso de la herramienta cut para crear los loops que necesitemos, pero hay que tener cuidado con ella, porque a pesar de que en teoría un corte pueda parecer

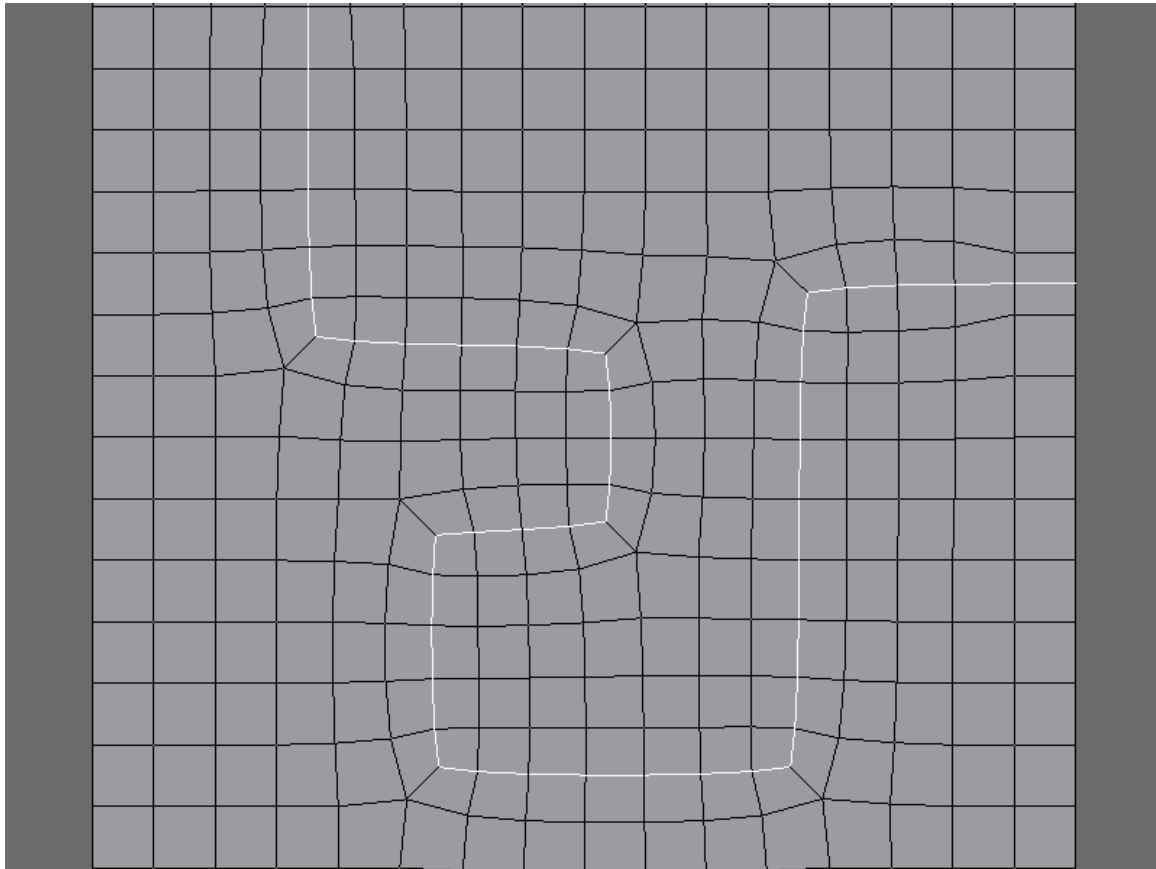


correcto, hay que corregirlo para que se adecue ya que esa primera aproximación puede ser errónea, como pasa en el siguiente caso:

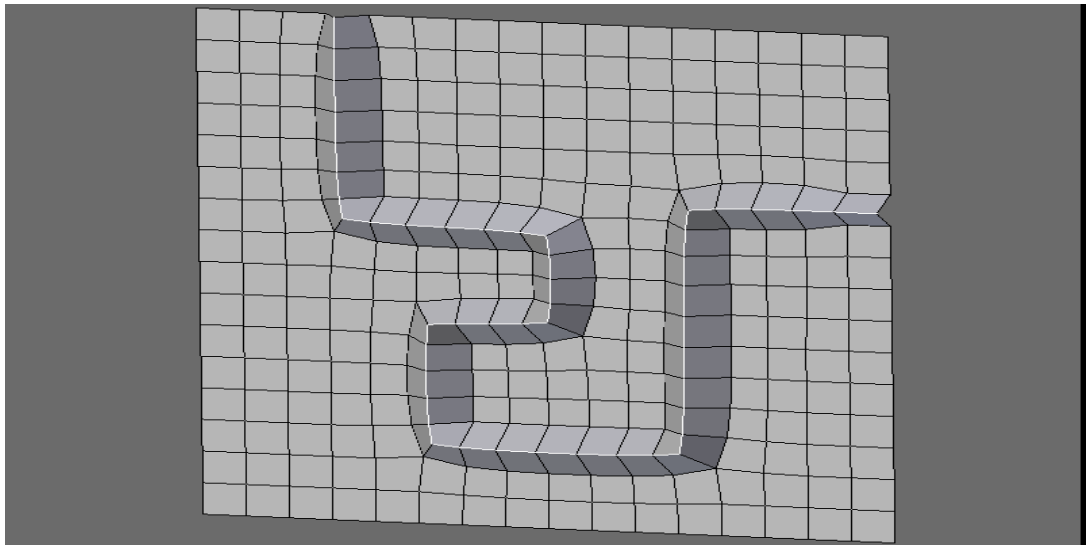
En un comienzo queremos crear el siguiente flow, por lo que usamos la herramienta cut sobre esa malla y siguiendo el siguiente patrón amarillo.



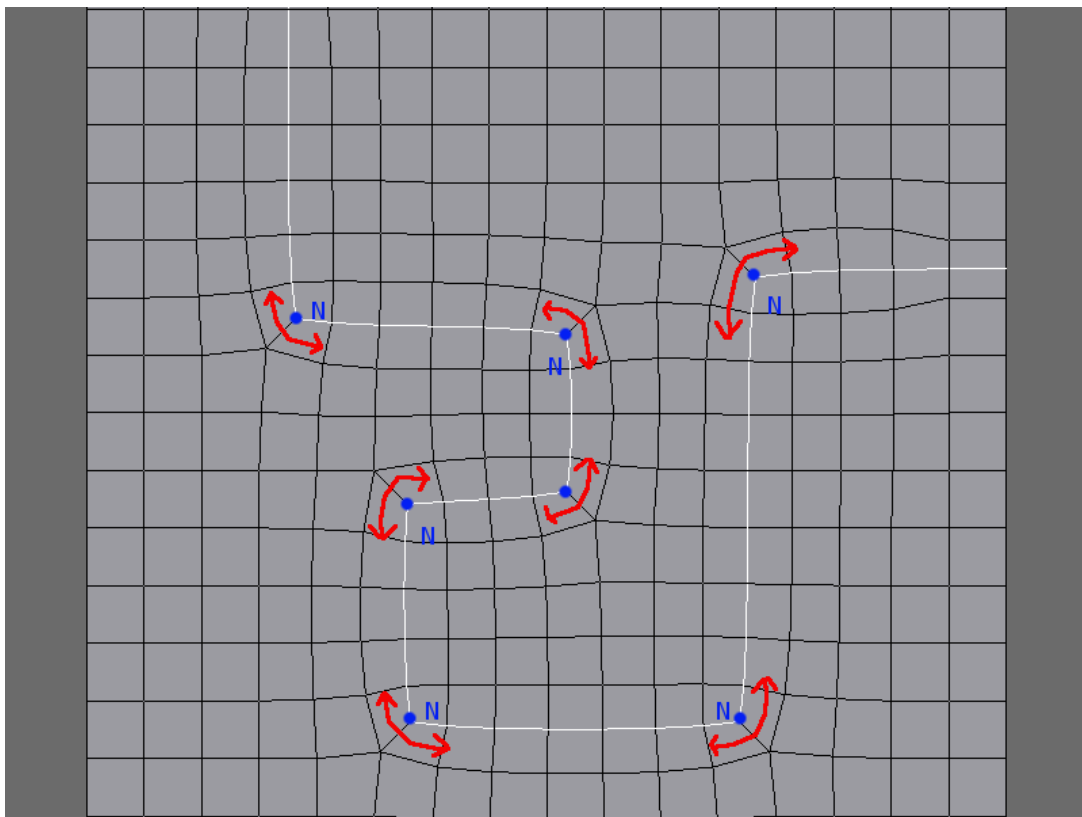
El resultado que nos da la herramienta cut es el siguiente, que a primera vista puede parecer correcto.



Si extruimos ese nuevo edgeloop, nos da un resultado como el siguiente.

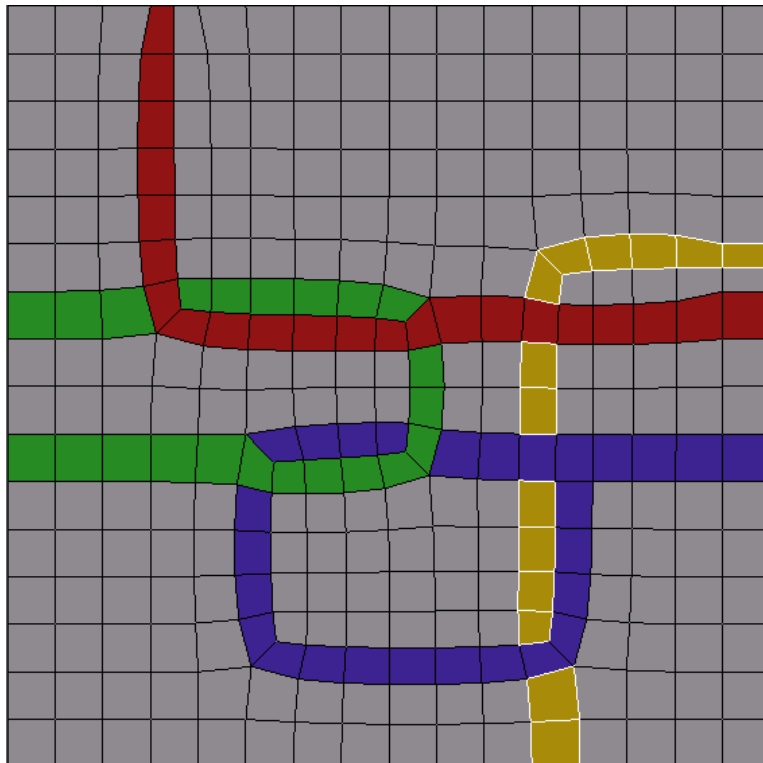


Si nos fijamos más en detalle, vemos que falla en cada una de las curvas ya que toma la curva sobre un V3, y las toma mal, por lo que se crean diferentes flows alternativos.

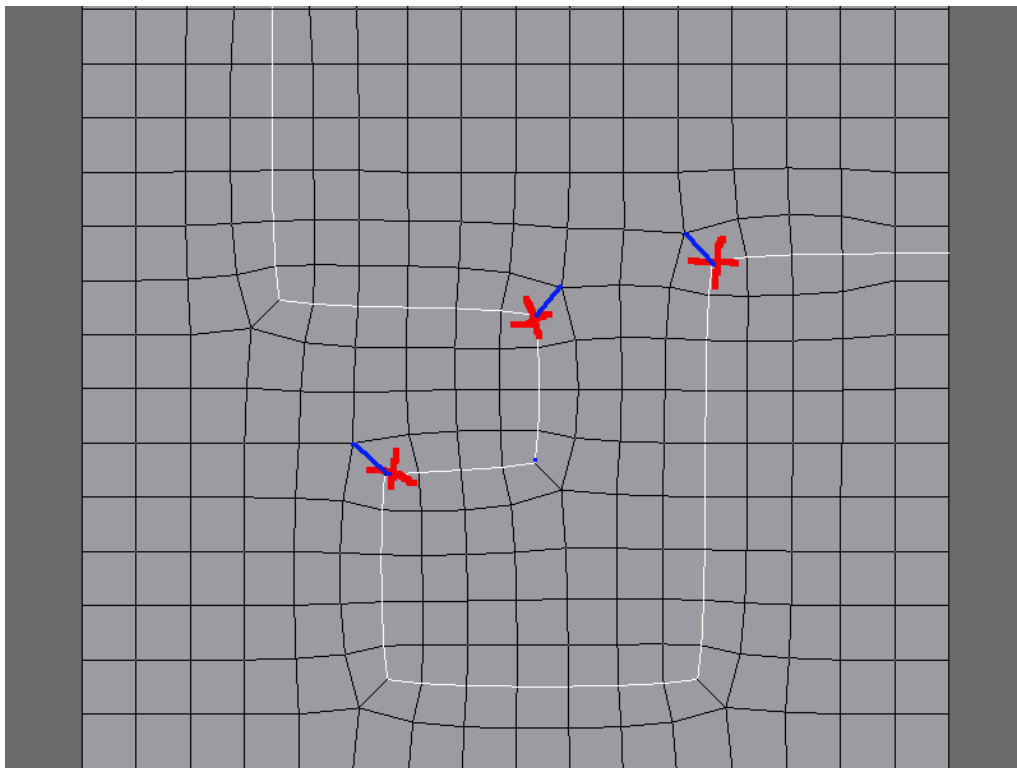


Si nos fijamos detenidamente, el flow gira alternativamente alrededor del pole. Eso hace que el resultado sean varios flujos en la superficie de la mesh. Al intentar usar ese edgeloop, vemos que se divide y se para en varios lugares. También observamos que no hay ningún problema si el loop tiene forma de C (C-loop), o se sigue en modo espiral, pero en el momento en que la curva tiene forma de S (S-loop), el formato de curvas falla y se descompensa. Por eso este problema no se suele dar en los loops usados por ejemplo

para los ojos cuando se hace box modeling, que es cuando más se usa la herramienta de cut, pero si se forma este problema al dar detalle. Si coloreamos los flows que se crean, salen los siguientes:



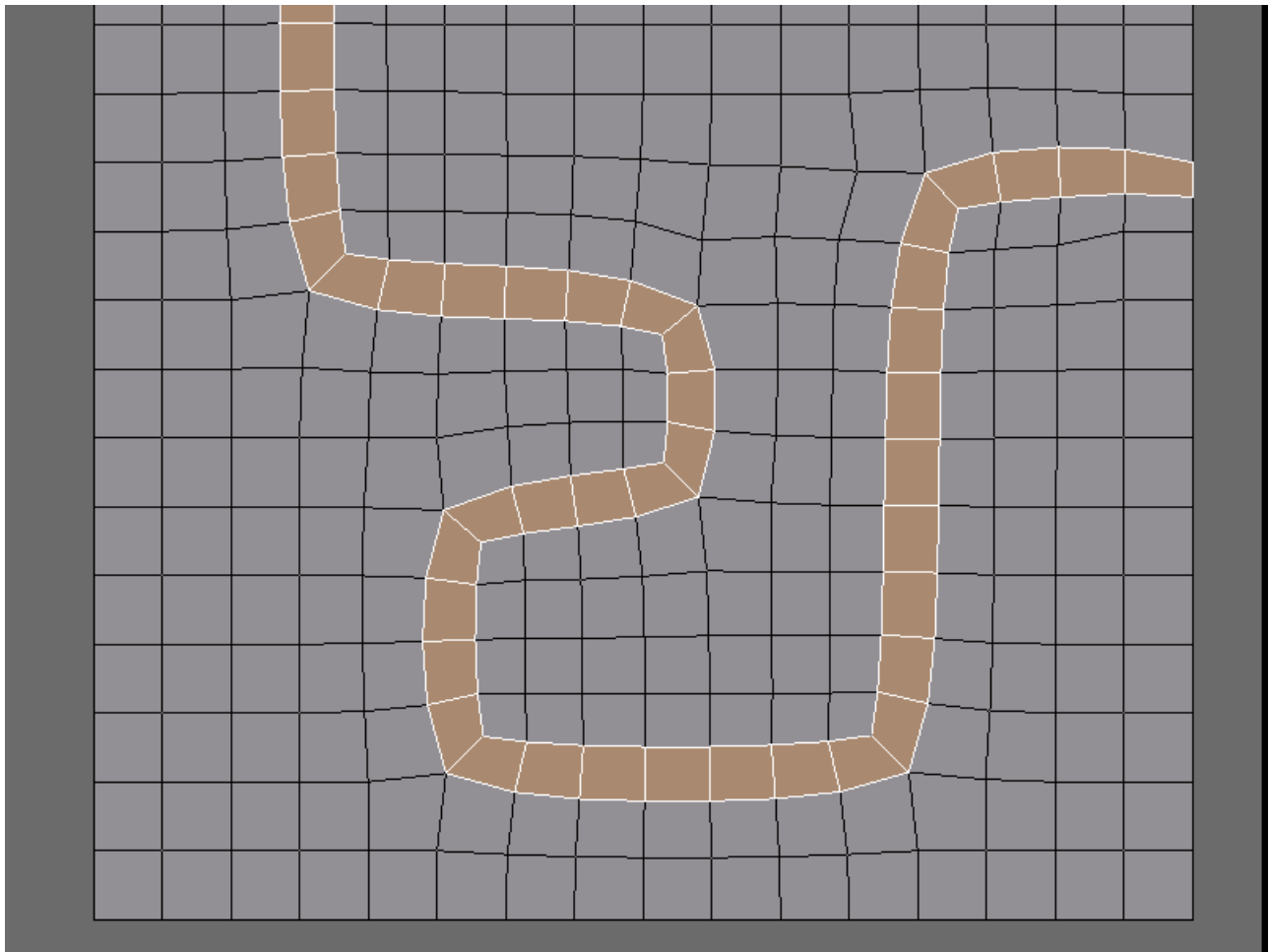
A pesar de ello hay una solución para este problema. Simplemente consiste en ver cuáles son los lugares donde hay problema



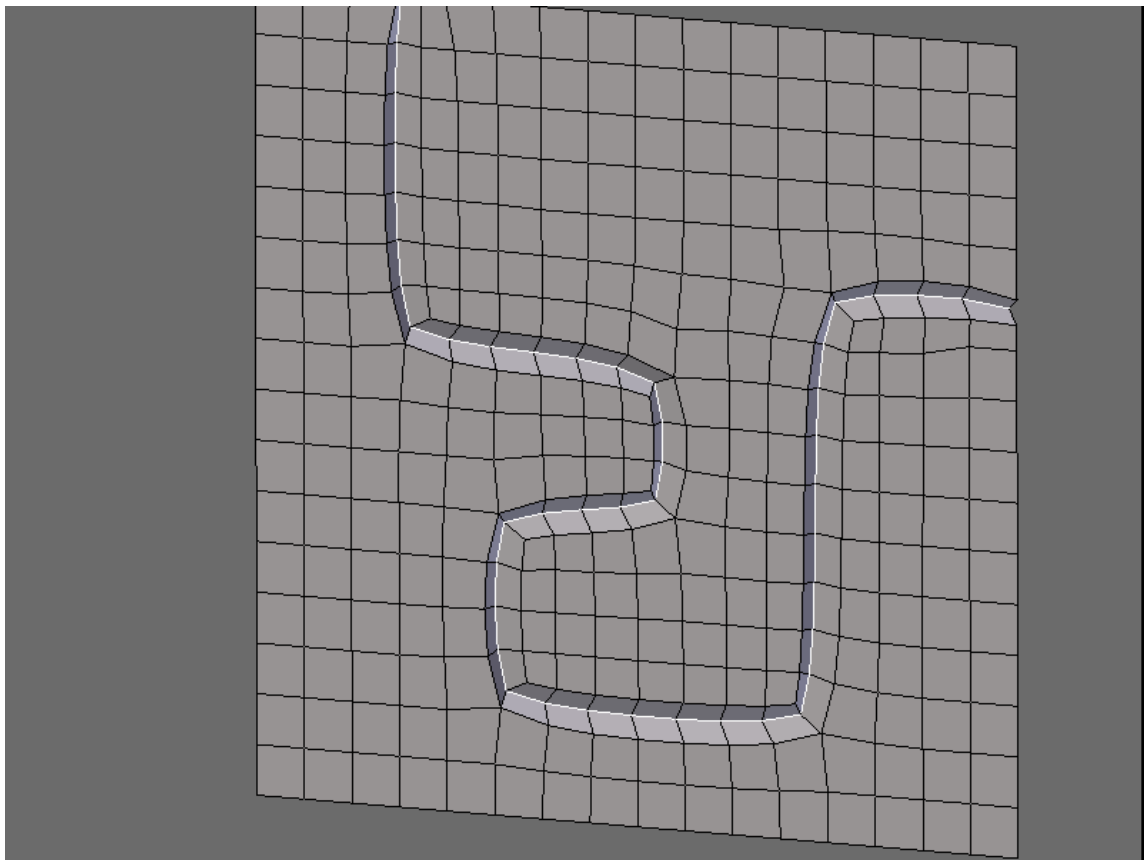
Los lugares señalados en azul son los problemáticos, pero simplemente hay que girarlos. Si el edge que va a ser girado fue precedido por un cambio de dirección de sentido antihorario, entonces el giro deberá ser en sentido horario. Por el contrario, si el cambio es de sentido horario, el cambio deberá ser en sentido antihorario.

Tras hacerlo habrá que borrar los edges y rellenar los diferentes polígonos como creamos conveniente.

El resultado final de todo esto es un faceloop simple.

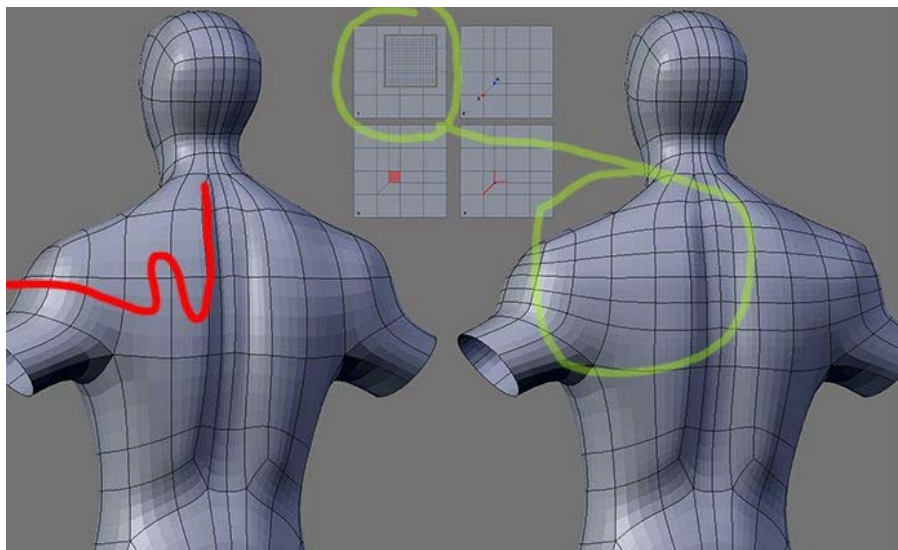


En el caso del principio, que queramos hacer una vena, simplemente con hacer un corte a lo largo de este faceloop nos dejará una topología de la malla limpia. Esto hace que a cada lado de último segmento creado hay un quad en cada uno de los lados, cosa que en el caso del principio no existía, ya que dos segmentos del mismo edgeloop compartían quads. Esto hacía que la malla se mostrase extraña.

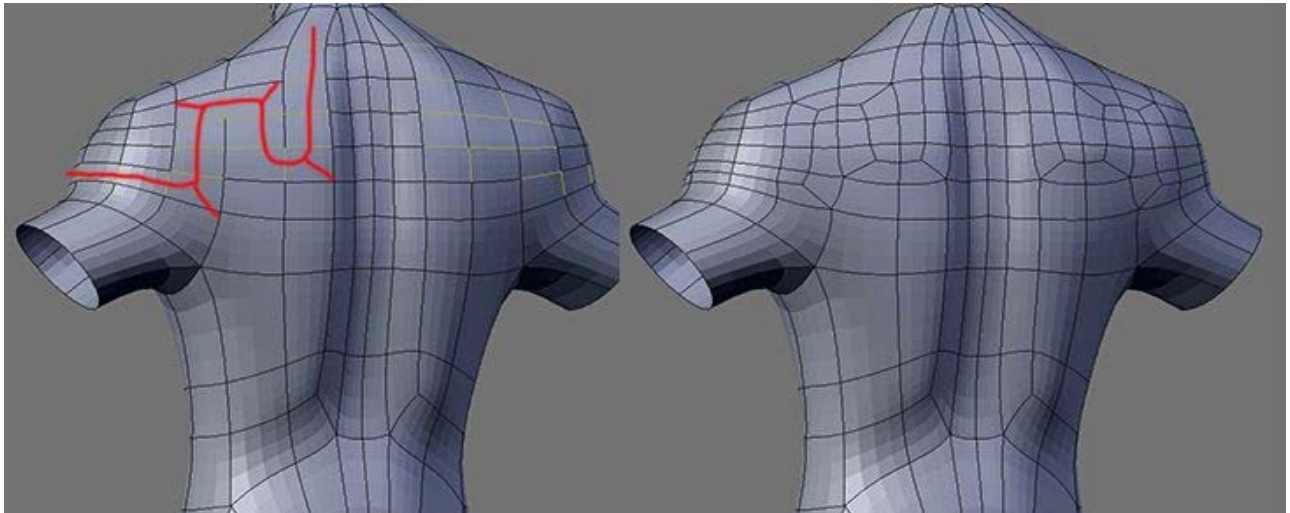


Teniendo todo esto en cuenta, tenemos que tener muy en cuenta el poder que nos proporciona los edgeloops en el proceso de modelado. Esto hace que de forma muy simple, podamos incorporar cualquier tipo de estructura que nos añada detalle a la malla de forma sencilla.

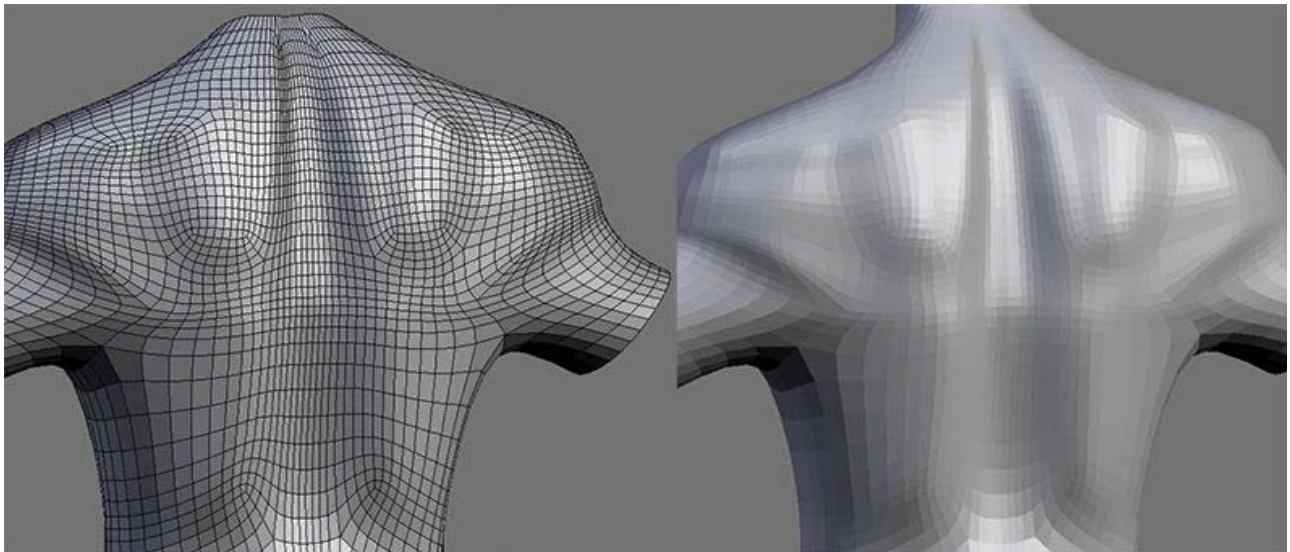
Pongamos el ejemplo de dar volumen a la espalda humana de forma simple. Es algo que a simple vista puede parecer laborioso. Pero usando lo que hemos desarrollado anteriormente, se puede hacer en un par de pasos.



El problema de eso es que en la imagen de la izquierda no hay suficiente geometría para poder realizar ese flow de manera correcta, por lo que para arreglarlo lo primero que se debe hacer es aumentar la cantidad de geometría que existe. Tras esto, simplemente con crear un keyloop siguiendo el flow que buscábamos anteriormente, se resuelve, aunque con ello no conseguimos aun todo el detalle que necesitábamos.



A pesar de que no se parecen al principio, añadiéndole un nivel de subdivisión vemos que con esas simples reglas podemos obtener una espalda con un buen detalle.

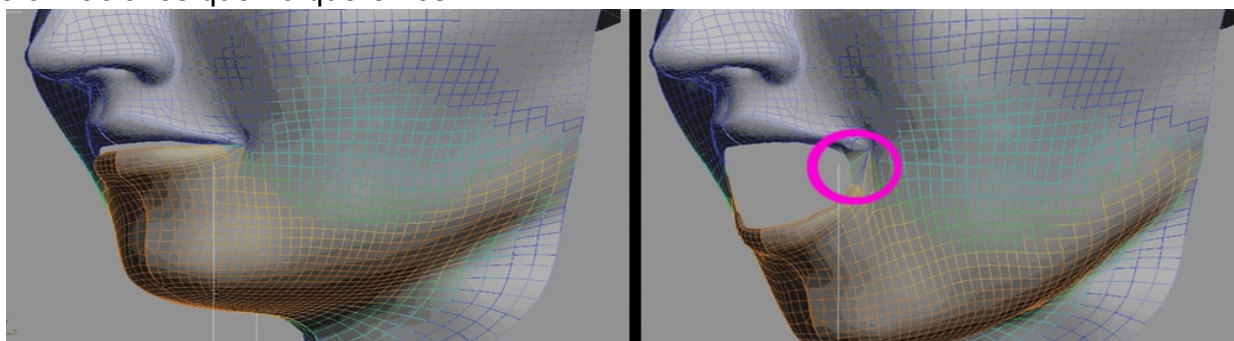


Hay que tener en cuenta también que no es una espalda anatómicamente perfecta, que era un simple ejercicio para ver el poder que se puede obtener gracias al buen uso de la técnica.

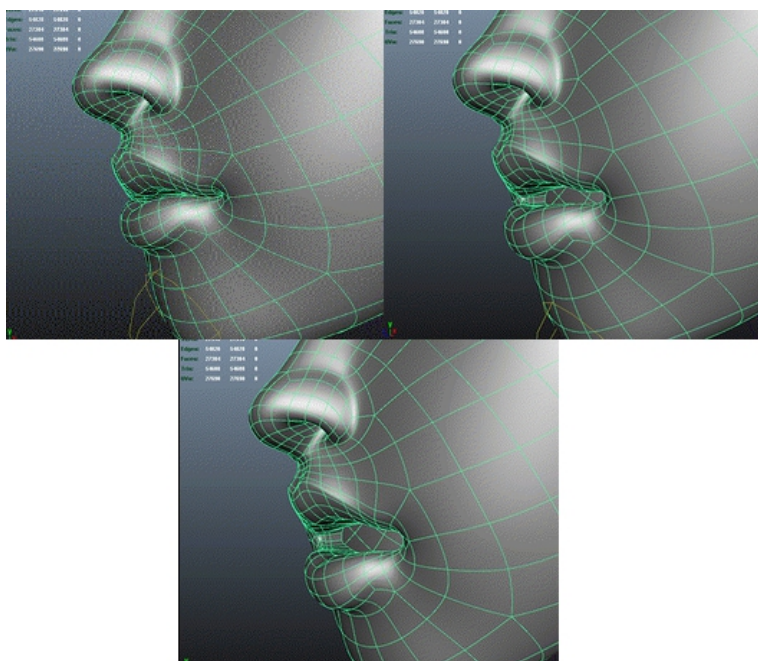
2.6 Modelado del rostro

En esta sección recogeremos todos los conceptos anteriores y los uniremos para crear el rostro humano. Crear un rostro humano creíble es una de las metas finales de la gran mayoría de artistas 3d. Para crear una cara realista y de apariencia natural hace falta una serie de pasos, desde recoger referencias, planear antes del modelado, modelarlo, texturizarlo, renderizarlo.... Y todo ello teniendo en cuenta que debemos crear un modelo pensando en poder animarlo posteriormente.

Teniendo muy en cuenta este último punto, tenemos que hacer uso de los edgeflows, que es la herramienta que más puede simular cómo se arruga y se estira el rostro. Por lo tanto el punto clave de la construcción de un rostro humano es la buena topología de la malla, ya que una mala topología provocará los temidos artefactos, pinzamientos y deformaciones que no queremos.



Una buena topología además nos ahorrará número de polígonos, además de la desaparición de todos esos problemas.

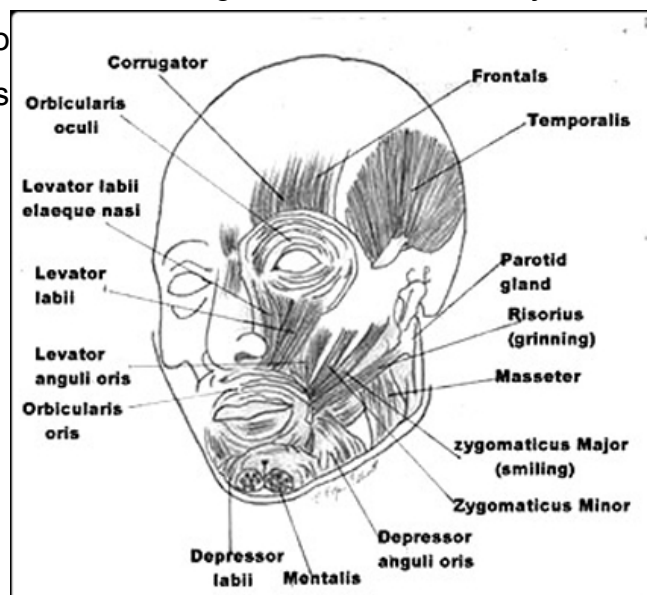


Lo primero de todo vamos a ver como es el rostro humano anatómicamente hablando.



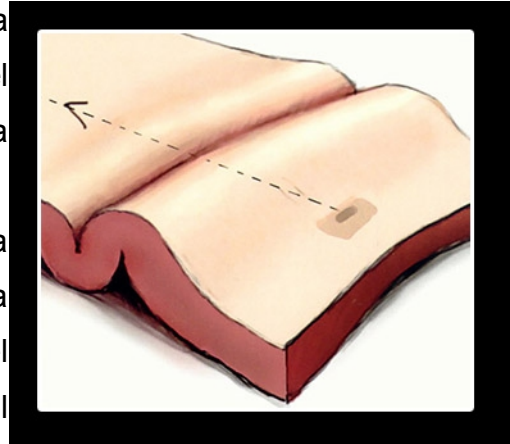
La forma de la cabeza humana, así como los detalles y las expresiones están formados por músculos y la forma del cráneo. A pesar de que el rostro humano tiene más de treinta músculos, y no todos unidos a hueso sino que hay algunos unidos a otros músculos, en cuestión de animación no hace falta tener en cuenta todos y cada uno de ellos. Hace falta tener en cuenta seis conjuntos y luego tener una visión global. Estos seis conjuntos son los que forman parte del conjunto nasolabial y de los ojos, que son las partes que crean la emoción.

- Orbicularis oculi
- Levator anguli oris
- Depressor anguli oris
- Levator Labii
- Risorious

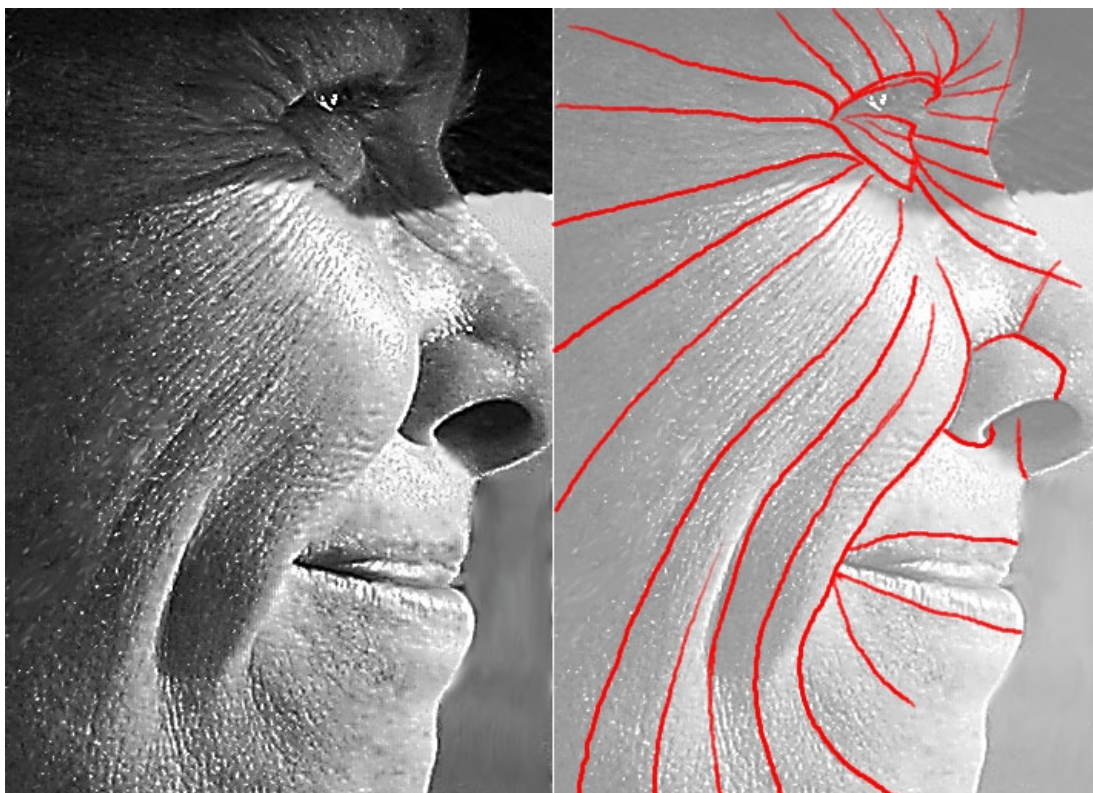


Estos músculos no crean la forma de la cara, pero son los que crearán las emociones, por lo que hace falta conocer su funcionamiento. Para ello la mejor forma es imaginarse una esterilla de goma encima de una mesa, unida con cuerdas por debajo. Cuando se estira una u otra cuerda, parte de la esterilla se desliza y se arruga. Esas cuerdas son invisibles a nuestra vista, pero lo que todos vemos es la marca que deja en la superficie. Para conseguir el efecto exacto debemos estudiar el exterior de la esterilla, e incluso la mesa, pero no las cuerdas.

La esterilla de goma se traslada en nuestro modelo a la superficie de la piel, las cuerdas los músculos, y la mesa el cráneo, por lo que debemos estudiar el movimiento de la forma de la piel exterior, no del músculo.



Por lo tanto la topología debe estar basada en las arrugas principales que son formadas por los músculos de la cara, por lo que una buena base de anatomía facial y de los músculos siempre es un bonus, pero es posible que el modelo no se vea realista si se estructuran todos los edges exactamente donde se encuentran todos y cada uno de los músculos y las arrugas, ya que no se acabaría jamás. Es mejor para nosotros exagerar esas arrugas principales de las imágenes de referencia y basarnos en ellas.



2.5.1 Topología

Dado que la gran mayoría de las caras humanas son similares al 99%, la topología de las caras no sufrirá muchos cambios. Aun así dependerá del estilo del artista, y de cómo se sienta más cómodo.

En la cara humana tenemos los siguientes elementos, cartílago (representado por el color azul), hueso (la zona blanca), músculos (representados por los colores rojos) y piel (que es parte del color amarillo). La piel no tiene movimiento si no es tensada por los músculos, por eso hacemos esa distinción.



En el caso de la animación, tenemos que trasladar todos esos elementos al modelo, y eso lo traduciremos en la disposición de los diferentes edgeflows. En el siguiente gráfico podemos ver los cuatro edgeflows principales de la cara.

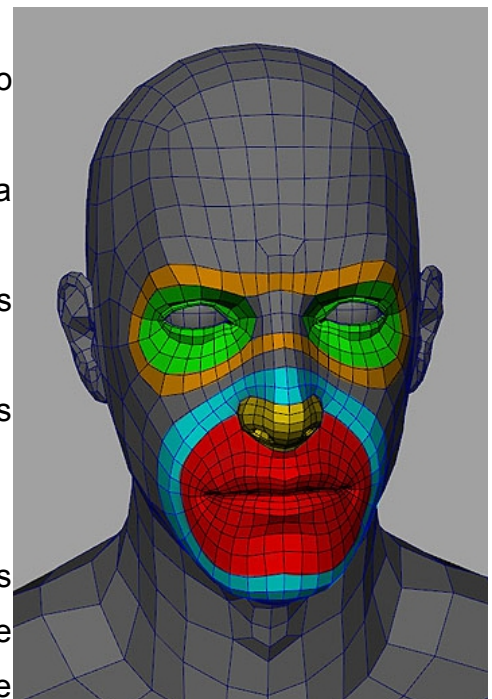
*Rojo: Zona labial, encargada del funcionamiento correcto de la boca y los labios.

*Azul: Zona nasolabial, encargada de la apertura de la boca y de la expresividad de la sonrisa.

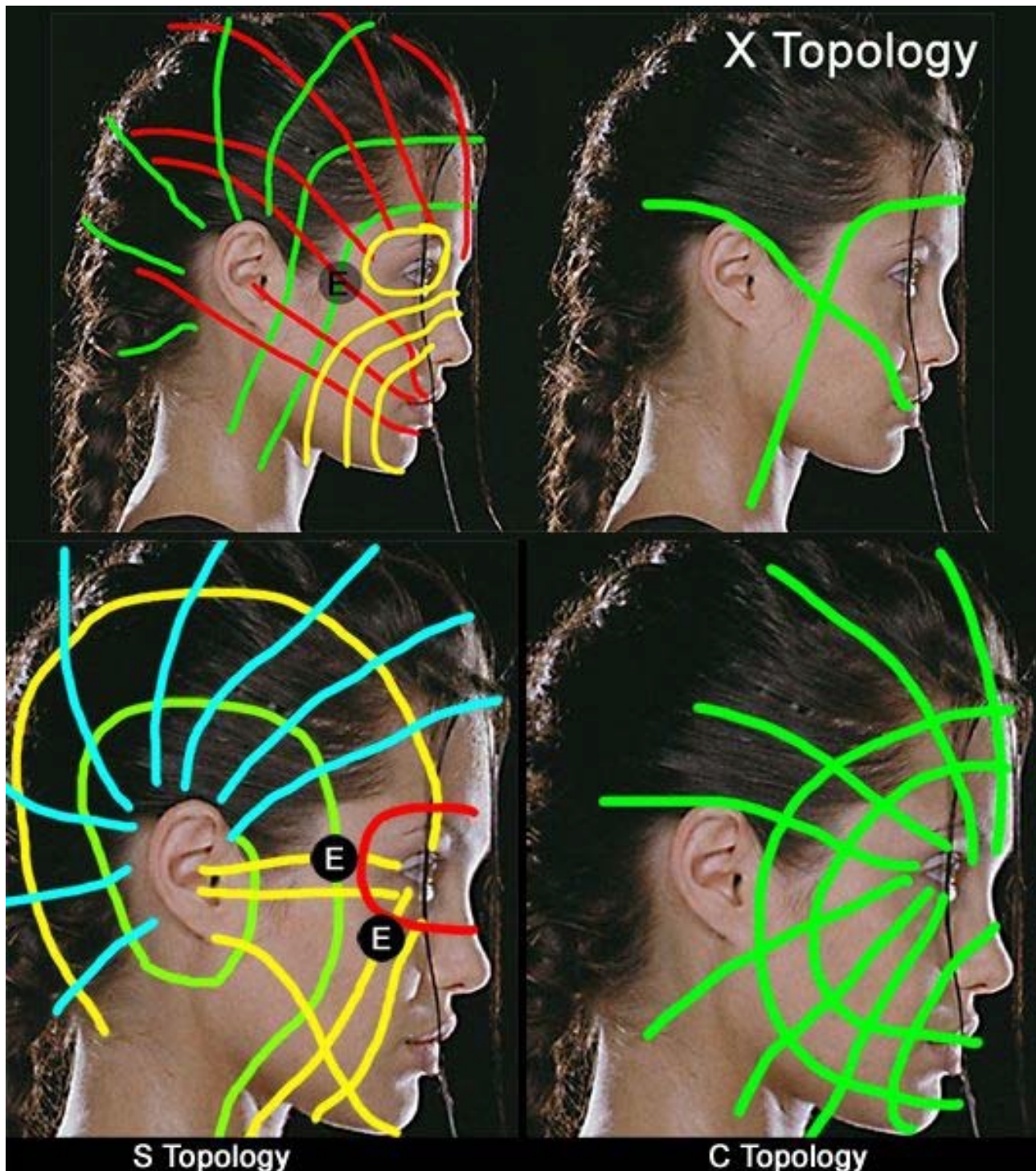
*Verde: Zona ocular, encargada de la apertura de los ojos.

*Naranja: Zona encargada de la expresividad de los ojos y cejas.

En algunos modelos también podemos tener otros dos edgeflows dependiendo de la morfología de la cara y de cómo queramos hacer el modelo. Podemos tener uno que delimite toda la mandíbula, y podemos tener otros para mayor expresividad en la frente, aunque estos funcionan de diferente modo al resto ya que son para añadir detalle.



Teniendo todo esto en cuenta, el artista puede ajustar los diferentes edgeflows a la morfología de la persona, qué es lo que quiere resaltar en el modelo, a qué tipo de animación está referido, a su forma de trabajar y a su experiencia. Según esto hay tres tipos de topología que se pueden formar en una misma cara, C, X y S.



C

Esta es la topología más sencilla de crear, y a su vez la más rápida. Por ello es muy

usada en la creación de personajes de estilo Cartoon o incluso criaturas. Es muy recomendable para personajes de estilo manga y es increíblemente rápida y eficiente.

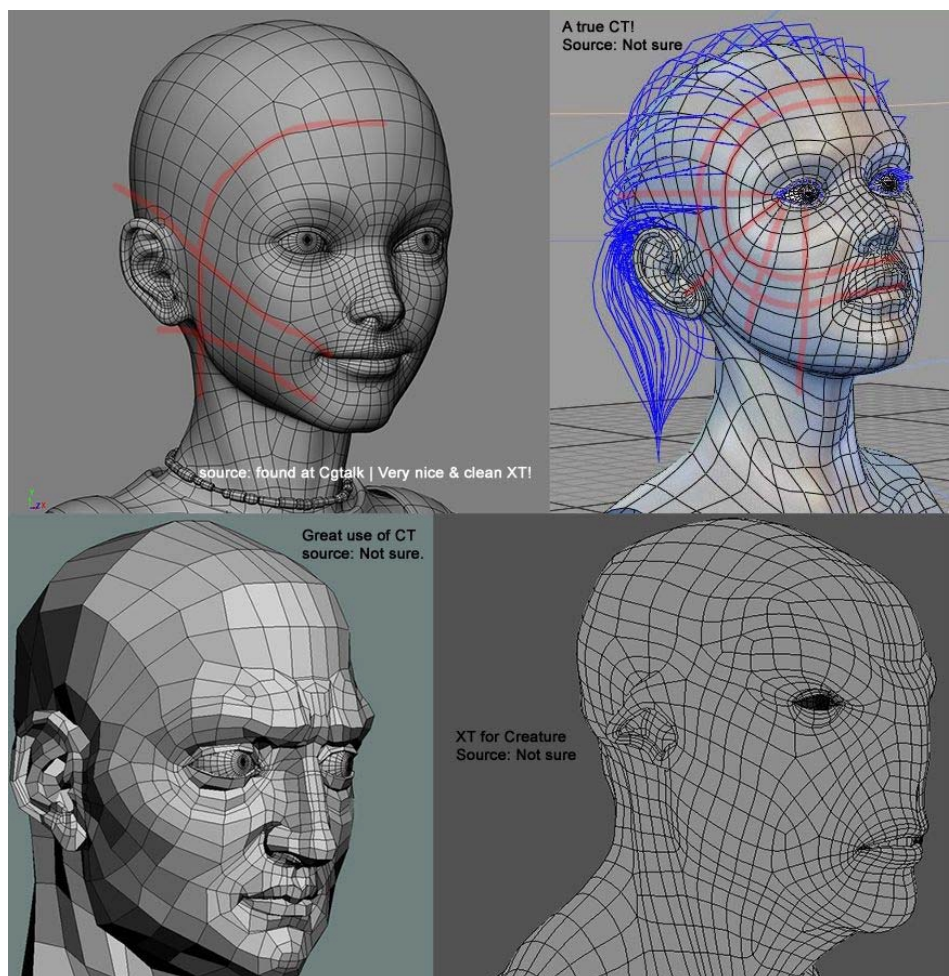
X

Usada en ámbitos profesionales, no se limita solo a la creación de modelos de estilo cartoon, sino que permite la creación de rostros realistas ya que tiene más control y define mejor todos los detalles y todos los aspectos necesarios en el realismo.

S

Esta topología es menos usada, ya que es la más complicada y también tiene ese efecto profesional, se puede combinar indistintamente con la X ya que cambia mayormente uno de los loops y lo lleva detrás de la oreja, mientras que la X lo lleva por la parte superior de la frente.

Estas tres topologías se pueden combinar entre ellas. Normalmente cuando se mira más de cerca, la topología C no tiene ningún pole en la mejilla o cerca, mientras que la X tiene normalmente un pole entre la oreja y el ojo. Por eso se dice que la disposición de los poles en la malla es lo que define la topología.



2.5.2 Pre-modelado

Antes de empezar el proceso de modelado, como hemos dicho, necesitamos referencias, y más si vamos a modelar una persona real. Para ello es importante tomar la fotografía con un ángulo correcto o nuestro modelo siempre lucirá mal y habrá que gastar más tiempo del necesario para poder reajustarlo. Además tenemos que tener en cuenta la distancia focal de la cámara, a mayor focal, mejor será la fotografía como referencia para el modelo. Esto se debe a que cuanto mayor sea ese valor, más se acercará al ángulo de vista ortogonal, como la vista frontal, o lateral, que son la base en la que modelamos.



A pesar de eso, la vista ortogonal provoca que el modelo tenga una ligera distorsión y se vea algo más gordo de lo que es, pero es algo fácilmente arreglable en la vista perspectiva. Por último tenemos que tener en cuenta la luz y tener una distribución

uniforme de la iluminación, ya que podemos perder detalles por sobreexposición o subexposición por exceso o falta de luz.

En mi caso, después de realizar varias tomas frontales, y de perfil, he elegido las que mejor cumplen para el objetivo final.

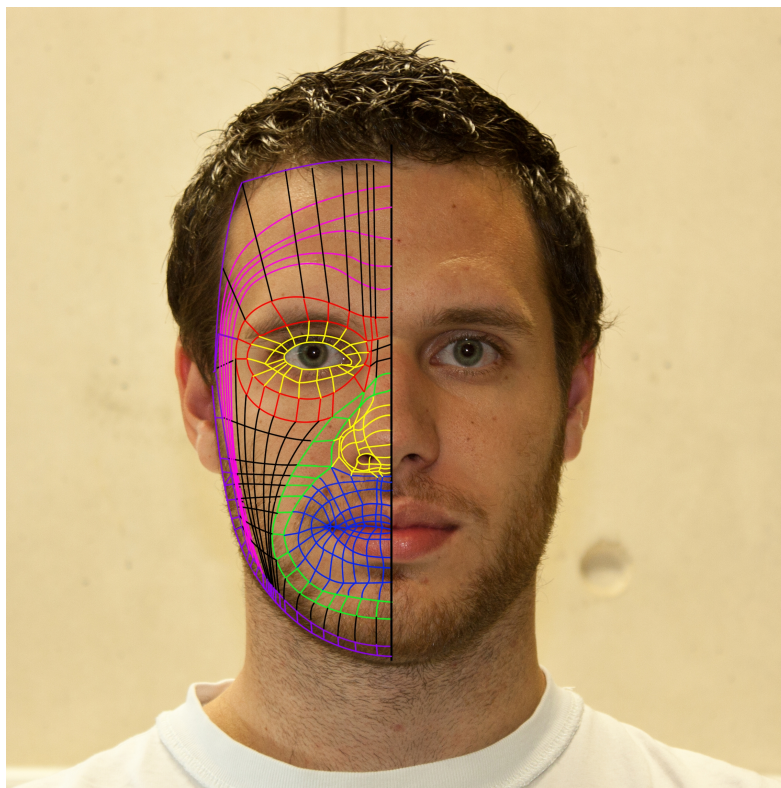


Así que después de tomarlas, han sido recortadas y se les ha hecho una corrección de color y de contraste, para poder sacar el máximo detalle de ellas y no perder nada. A su vez se han alineado para que los diferentes detalles casen entre sí, y tengan el mismo tamaño, ya que nos servirán para referencia posterior en el modelado.



Tras esto, ya tenemos las imágenes de referencia, por lo que hay que seguir al siguiente paso. En él dibujaremos la topología que vamos a seguir en el modelo sobre la propia foto. Esto nos ayudará en un futuro para poder, primero localizar dónde deben ir y por qué se

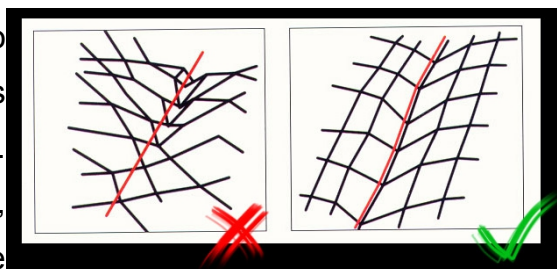
situa ahí cada uno de los diferentes edgeloops, y todo ello teniendo en cuenta la morfología propia de cada cara con las arrugas y pliegues característicos de cada persona, como hemos visto antes. Al ser un modelo fotorealista, escogeremos la topología en X, ya que es la que tiene un acabado más profesional para lo que nos interesa.



Hay varios loops que son claves, como hemos visto en la sección de músculos, y que determinan esta morfología. El loop principal, por el cual he estructurado más o menos el resto de loops, sería el nasolabial, un loop que nos permite el poder abrir y cerrar la mandíbula, dibujado con el color verde. El siguiente loop clave es el que tiene forma de antifaz alrededor de las cuencas de los ojos, de color rojo. Este nos permite delimitar al zona de los ojos, así como situar los primeros poles que bifurcarán los diferentes loops entre el verde y el rojo, además de que en la parte de animación es el que da vida a la zona de las mejillas y de las cejas. El siguiente loop es el morado, que determinará la forma de la cara ya que delimita la mandíbula y sube hasta la frente hasta el comienzo del pelo. Luego está el azul, encargado de la expresión de la boca y de los labios, contenido en el loop verde. En la separación entre las dos zonas, nos encontramos otro de los poles principales, que nos dará paso a un loop que definirá la nariz. Después está uno de los más obvios, que es el de los ojos, también pintado de amarillo, situado dentro del antifaz rojo. Y por último, aunque no es loop como tal, se podría englobar dentro de ellos ya que es lo que da la forma en X a la topología general, que son los edgeloops que van desde la

frente hasta el cuello.

Después del proceso de pensar dónde deben ir los diferentes loops, tenemos que tener una última serie de cosas en mente cuando creamos nuestro modelo. Lo primero, debemos intentar que sea de la mínima resolución posible. A su vez debemos usar los quads todo lo posible, ya que los triángulos no se deformarán cuando se



animen, pero en el caso que no podamos evitarlos fácilmente, será mejor dejarlos que arreglarlos a cualquier precio, ya que en muchas de esas medidas se añadirán una gran cantidad de polígonos innecesarios. Se deben evitar los poles, o sino evitarlos sí intentar ocultarlos en zonas que no se vean mucho y no interfieran con la animación, además hay áreas donde los poles se deben situar para poder delimitar los loops. Los polígonos deben ser lo más regulares o cuadrados posibles, no deben ser estrechos o de forma de diamante, quitando alguna excepción que se pueda dar para formar arrugas. La dirección de los edges no deben ir en contra a la forma de la superficie o de la expresión.

Por último hace falta preparar el área de trabajo. Definiremos las unidades en las que estamos trabajando, y pondremos una cinta que nos indique precisamente las medidas deseadas. Por último colocamos las fotografías de referencia en un par de planos y congelarlas.



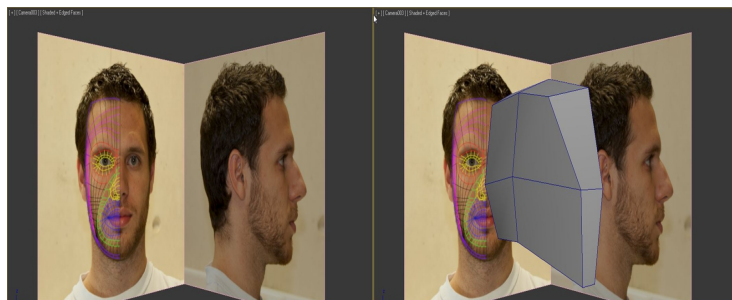
Dicho esto, nos disponemos a modelar la cabeza con los dos métodos escogidos, edge modeling y box modeling, ya que son los que más ventajas nos ofrecen y los más usados, para poder compararlos los dos.

2.5.3 Box modeling

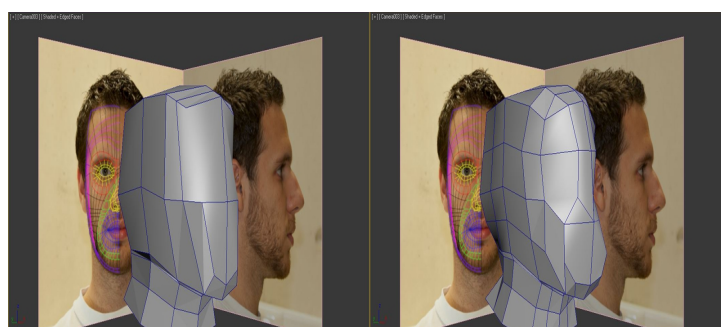
La técnica de box modeling, como ya la hemos descrito anteriormente, consiste en desde una primitiva simple, normalmente una caja, ir esculpiéndolo y añadiendo los diferentes elementos consiguiendo la forma deseada.

En nuestro caso empezaremos con una caja como primitiva principal. Para realizar la mitad del trabajo, retiramos uno de los lados de la caja y le aplicaremos el modificador symmetry, nos beneficiamos de que el cuerpo humano tiene dos mitades casi idénticas y así

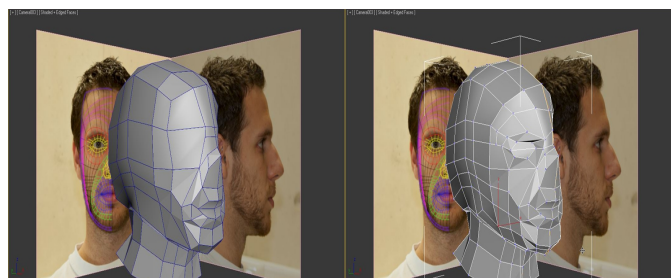
nos ahorramos tener que hacer el mismo trabajo en el otro lado de la cara. En esa caja tendremos una serie de subdivisiones iniciales, para poder darle la forma inicial de una cabeza.



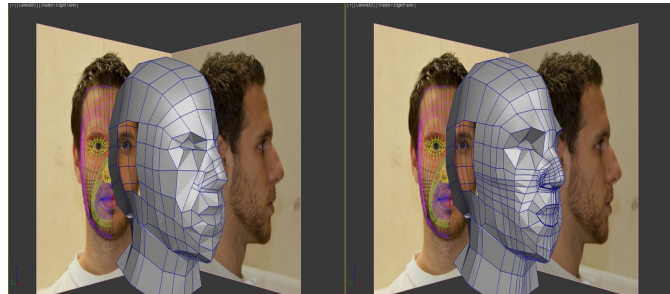
Tras esto extruimos para la formación de lo que sería la cara, y volvemos a extruir para crear el cuello. Con cada paso añadimos más y más geometría, que nos permitirá ir añadiendo detalle progresivamente. En estos primeros pasos no nos preocuparemos del número de lados de las diferentes caras, ni si son topológicamente correctas esas caras para una posterior subdivisión. En estas fases lo que importa es ir creando la forma final que tendrá, e ir situando poco a poco los poles y dónde se encontrarán las diferentes partes del rostro.



Por eso iremos ya situando y creado la zona superior e inferior. Esas dos zonas se convertirán en los lugares donde se encontrarán los ojos y la zona de la boca. Para ir creando nueva geometría usamos las herramientas de connect y de cut, con lo que podremos ir añadiendo nuevos edges donde los necesitemos. Tras crear las zonas, empezaremos a crear los primeros flows según la plantilla, e iremos situando poco a poco los diferentes poles. El primer flow que haremos será el nasolabial, el verde en la foto. Con él ya podemos estructurar dónde se situarán las zonas principales.



Podremos ya empezar a sacar la zona de la nariz, así como situar los la parte de los labios. Al situar la zona de los labios pondremos uno de los poles principales, el que hace de bifurcación entre la zona azul de los labios, la zona amarilla de la nariz, y la zona verde del flow nasolabial. Por lo que ya podemos empezar a añadir detalle y a hacer la forma de la boca poco a poco. Además, podemos ya crear, en el hueco que habíamos dejado antes, la primera aproximación de la zona de los ojos.



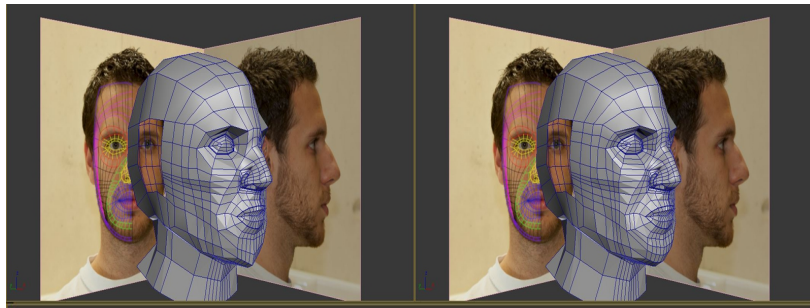
Tras ello iremos a por los dos últimos flows de la cara, que es el que delimita la mandíbula, y luego el de la nariz. Además crearemos el hueco donde se situará la oreja. La oreja es uno de los elementos más complicados, por lo que en muchas ocasiones es recomendable crearla a parte. Este procedimiento lo seguiremos usando en el proceso de modelado por edge modeling, así no hará falta crear dos orejas diferentes, y sólo con incluirla y soldarla al resto del modelo basta.

A partir de ahí es ya empezar a añadir geometría, en este caso hemos empezado a definir la nariz, creando el pole que manda el flow hacia el agujero de la nariz, con lo que podemos añadir más edgeloops sin masificar el resto de la malla innecesariamente. Eso es algo que hay que tener muy presente durante todo el proceso, hay que minimizar el número de edgeloops, y si acaso arreglárselas para cerrarlos donde sea, ya sea mandándolos a la otra parte de la cara con el symmetry, o a sitios donde se puedan concentrar a través de poles. En el caso que queramos añadir edgeloops a alguno ya existente, podemos usar alguna herramienta como chamfer, que nos creará más geometría fácilmente. Además, también pensando en la subdivisión posterior, hay que reordar que entre más cerca estén los edgeflows entre ellos, más se marcará la forma que queremos buscar, por eso se encuentra una mayor concentración de edgeflows cuando queremos marcar una arruga, o una cosa algo más brusca.

El caso más claro que se puede ver en este caso es la delimitación del contorno de los labios, que al poner un par de loops muy cercanos, se marcarán bien y se distinguirán del resto.

Antes de pasar a la última zona importante, añadiremos algo más de geometría en la

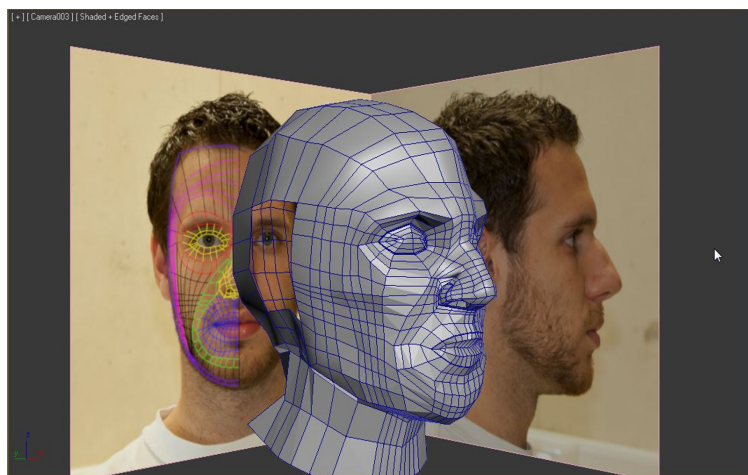
frente siguiendo el patrón en X para darle más forma al cráneo y que no tenga tanta forma de caja.



La siguiente zona que debemos retocar serán los ojos y la parte de las cejas. En este caso hemos creado los ojos cerrados, ya que en el siguiente caso los hemos hecho abiertos y así poder ver la diferencia al crear unos y otros. Hay que tener una cosa en cuenta antes de nada, y esto también se aplica a la boca. De la parte superior debe haber el mismo número de loops saliendo que de la parte inferior, sea de la boca como de los ojos, y deben de coincidir la posición de estos, ya que cuando se anime si no es así se verá algo raro y dará una sensación antinatural.

Para poder crear correctamente los ojos hace falta que tengamos una referencia, por lo que la mejor opción es usar una esfera para poder darle el contorno deseado.

El resultado final después de mejorar la frente y algunas partes que necesitaban algo más de geometría es este.

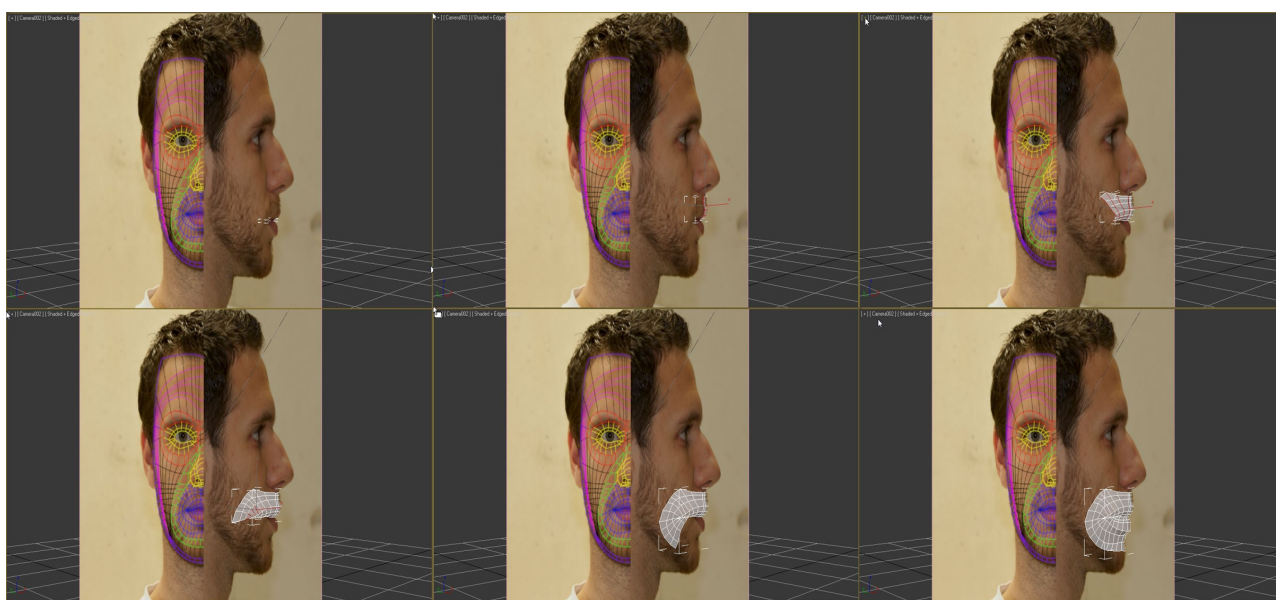


Posteriormente se puede añadir mucho más detalle añadiendo edgeloops y modificando los existentes para sacarle todo el partido.

2.5.4 Edge modeling

Esta es la otra modalidad que vamos a desarrollar para formar nuestra cara. Consiste en empezar desde una primitiva como un plano, o desde una curva, y de ahí ir extruyendo las diferentes caras para darle la forma. Es una forma más precisa que la anterior, pero mucho más lenta. Además en esta es muy importante etapa de planificación anterior en la que localizamos los diferentes flows en la cara, ya que nos puede servir de plantilla. En esta metodología lo complicado es darle la forma correcta, ya que no tienes una vision global del modelo.

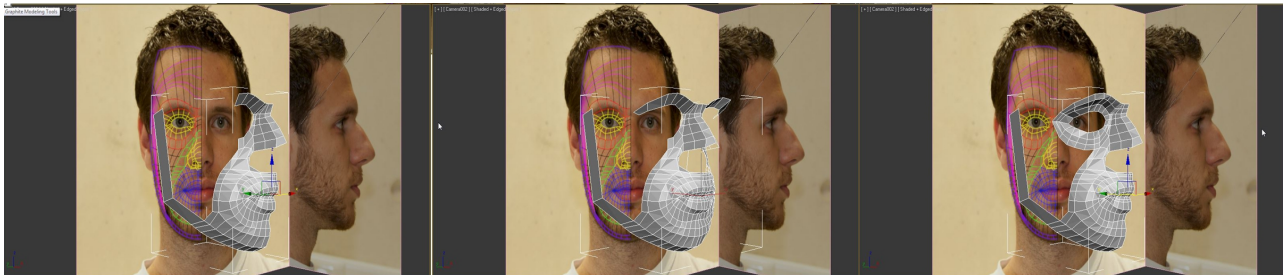
En nuestro caso empezaremos con un plano, e iremos siguiendo fielmente la plantilla que habíamos dibujado en la foto. Empezaremos desde la boca, toda la zona de los labios.



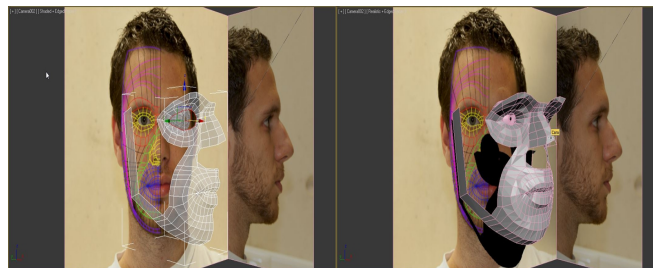
Con todo este proceso iremos controlando también en la medida de lo posible la profundidad de cada vertice y de cada edge colocado, ya que por lo general si se hace bien, no hace falta muchos retoques masivos posteriores. Una herramienta muy util es el poder seleccionar los vertices con soft selection, con lo que los cambios que hagamos no serán muy bruscos y se verá una progresión entre ellos, y así nos libramos de tener aristas y cosas extrañas. A su vez, como hemos visto antes, en el caso que queramos resaltar alguna zona, añadiremos algo más de geometría.



De ahí vamos creciendo poco a poco y creamos los flows adyacentes, que es el que delimita la zona de la nariz y de los pómulos, y luego uno que será el que nos haga de guía de por dónde irá la mandíbula. Además, como luego una gran cantidad de edgeloops tienen que ir ahí, ya lo tenemos como referencia y simplemente los tendremos que unir a ese primer segmento.



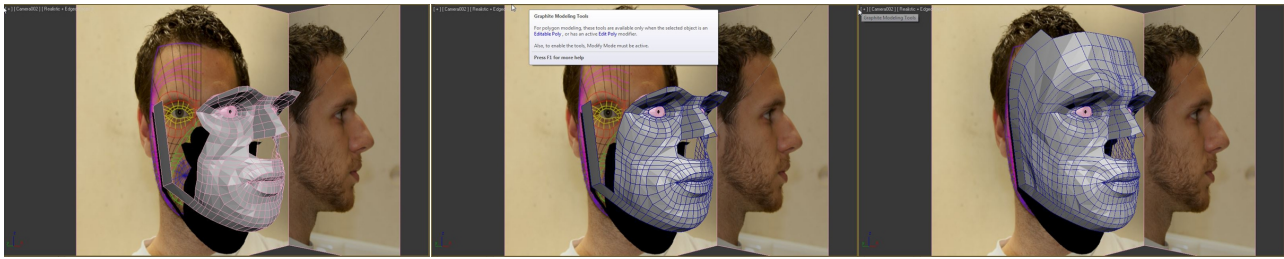
Tras ello desarrollaremos el tabique nasal, que nos dará paso a poder desarrollar el entrecejo, y el flow que delimita las cejas y las mejillas. Con eso podremos tener ya creado la gran mayoría de edgeloops importantes para la animación. Eso hace ya empecemos a tener una vision algo más global de la cara, con sus proporciones, por lo que ya a partir de este paso iremos corrigiendo poco a poco la forma total del modelo.



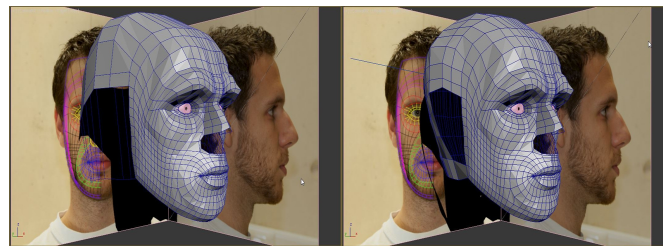
Los siguientes pasos es empezar a crear la forma de los ojos. Para ello, como hicimos en el anterior proceso, crearemos unas esferas que nos sirvan de referencia. Como el ojo lo vamos a crear abierto, en este caso nos interesa darle realismo al ojo. El ojo no debe ser una esfera perfecta, ya que se vería como una muñeca y muy artificial. Por lo contrario, el ojo humano es algo más achatado en la parte visible.

Así que para modelarlo debemos achatar una esfera, y dividir el ojo en dos partes, la parte del globo ocular, y la parte del cristalino. En la parte del globo modelaremos el iris, y la pupila. Y luego vendra otra parte que será la parte del cristalino, que cubrirá el iris.

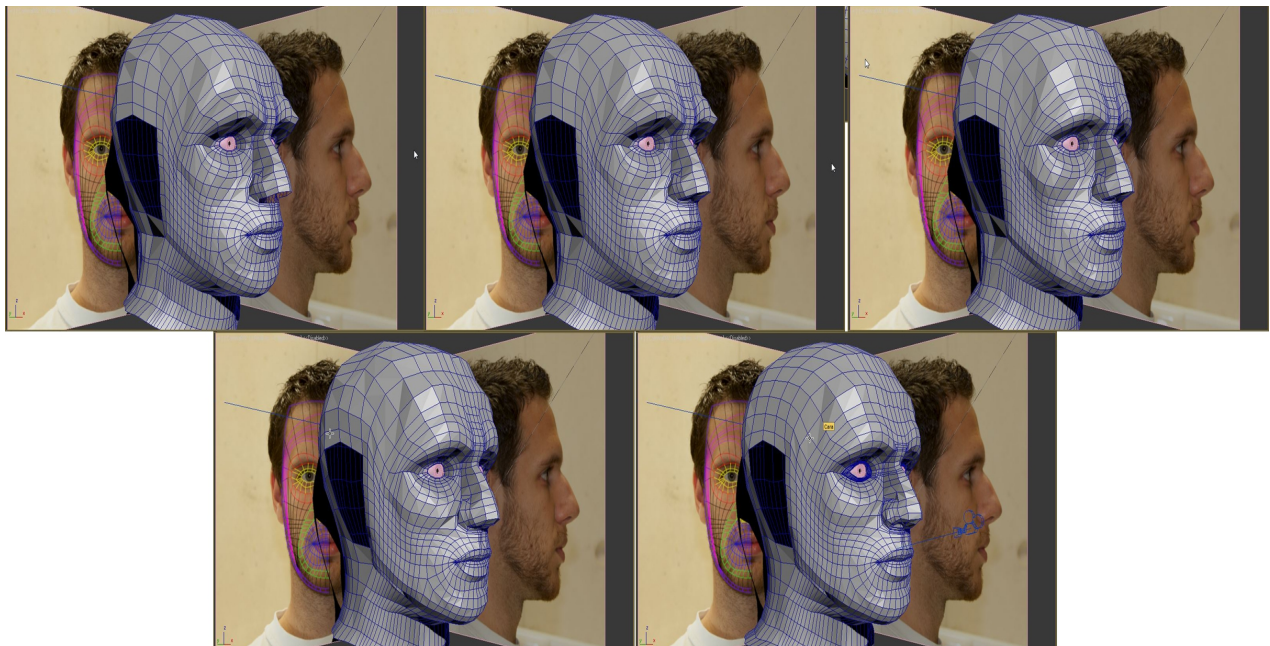
Una vez que tenemos la esfera creada y situada en el punto donde nos interesa, crearemos la piel del modelo que serán los párpados.



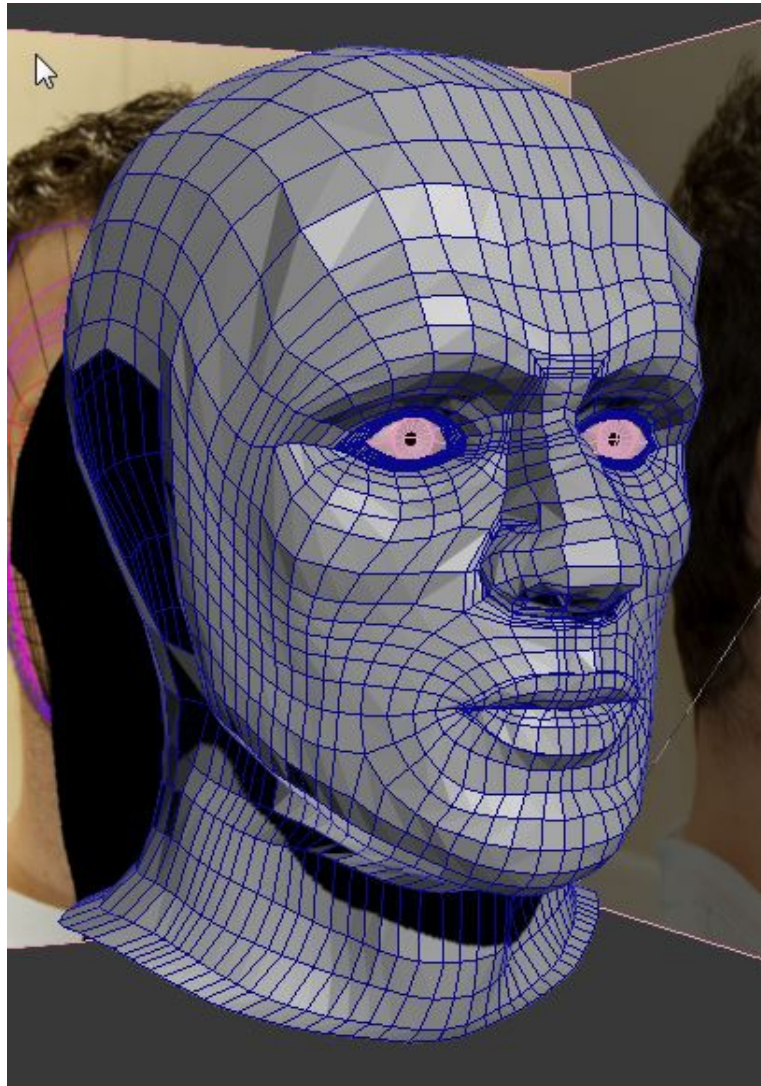
Los siguientes pasos son para ir formando el resto de la cara, desde los lados laterales a toda la parte de la frente, y uniéndolos al flow que forma la mandíbula. Con ello tendríamos hecho el rostro.



Los siguientes pasos son para completar el cráneo y ya tener todo preparado para los últimos pasos. Este es el momento en el que se ve ya por fin la forma completa, por lo que aquí es cuando se empieza a retocar en serio las diferentes zonas para que queden proporcionadas.

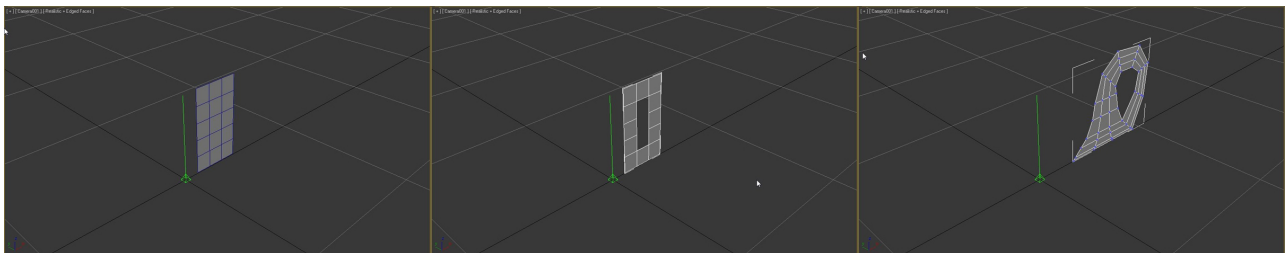


Por último modelamos el cuello de forma básica, y situamos el ultimo pole que nos permitirá crear la nariz. Para finalizar añadiremos geometría a los párpados y mejorar la forma final de todas las partes, además de intentar desplazar entre ellos para que no haya grandes diferencias de tamaño. Por último se da algo de forma al cuello, aunque no tocamos la geometría, ya la retocaremos más adelante.

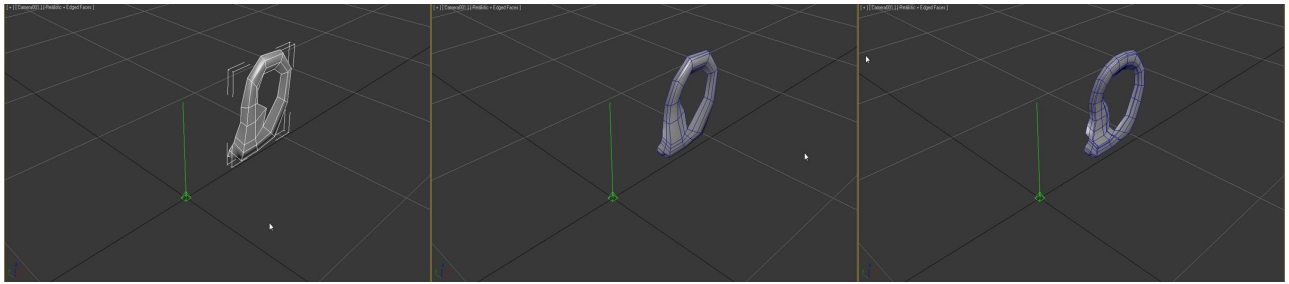


2.5.5 Creación de la oreja

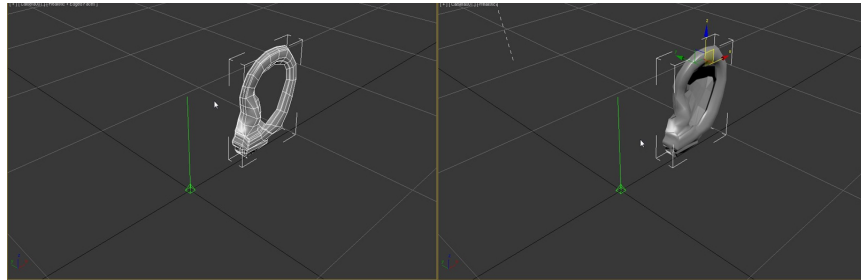
Para la creación de la oreja necesitamos empezar desde un plano 5x3.



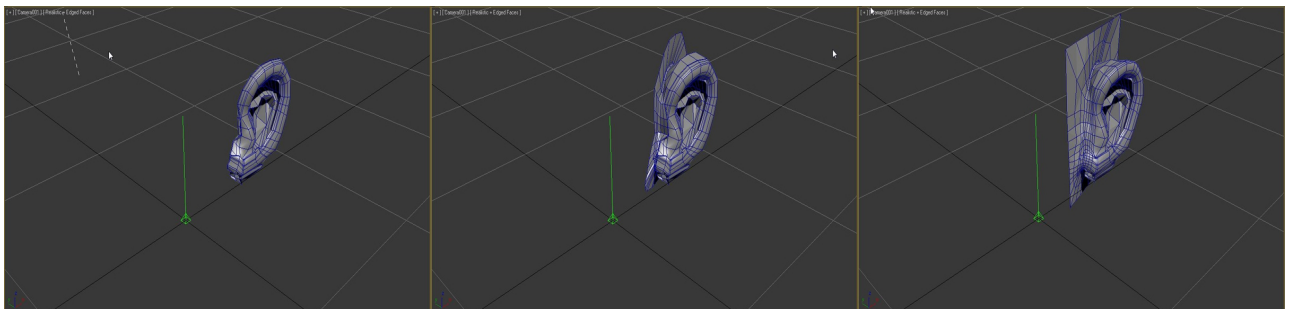
Para que nos sea más fácil hemos medido cuánto es el hueco que hemos dejado en nuestro modelo para alojar la oreja, así podremos soldarlo cuando toque más fácilmente. De ese plano eliminaremos los polígonos centrales, y le daremos forma espiral, ya que nos servirá como base para el resto de la oreja.



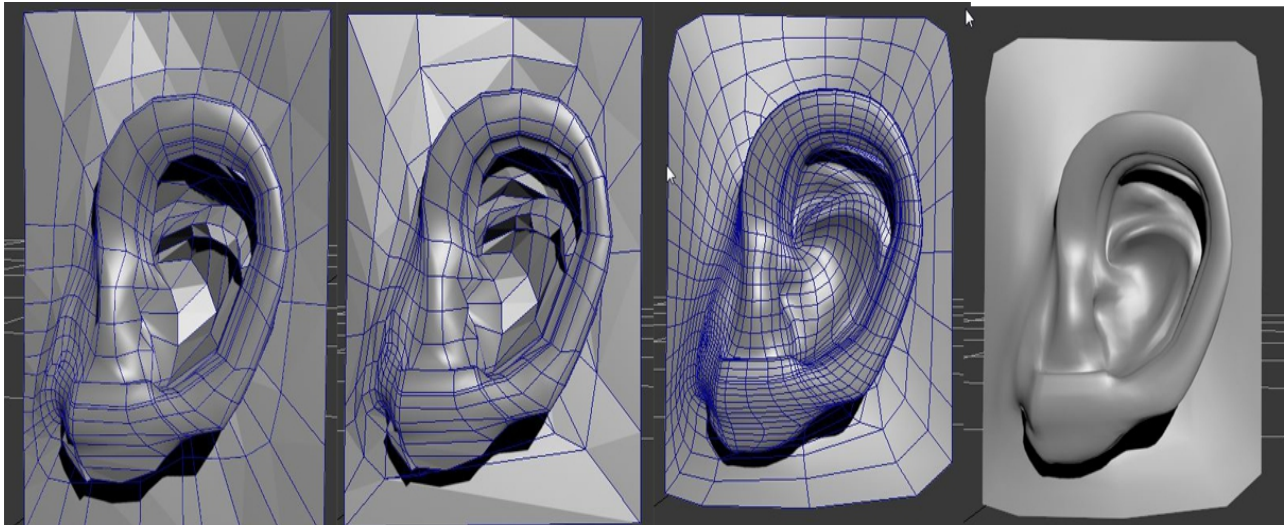
Poco a poco iremos dando forma a la parte exterior de la oreja, para poder preparar el terreno para la parte interior. Iremos añadiendo geometría para modelar mejor todo el borde, así como para poder hacer el lóbulo.



Prolongaremos los edges de la parte media de la oreja y seguiremos creando una espiral, con lo que nos proporcionará suficiente geometría para tener la base de todo el pabellón. A su vez vamos dándole profundidad a la oreja para hacer el hueco.



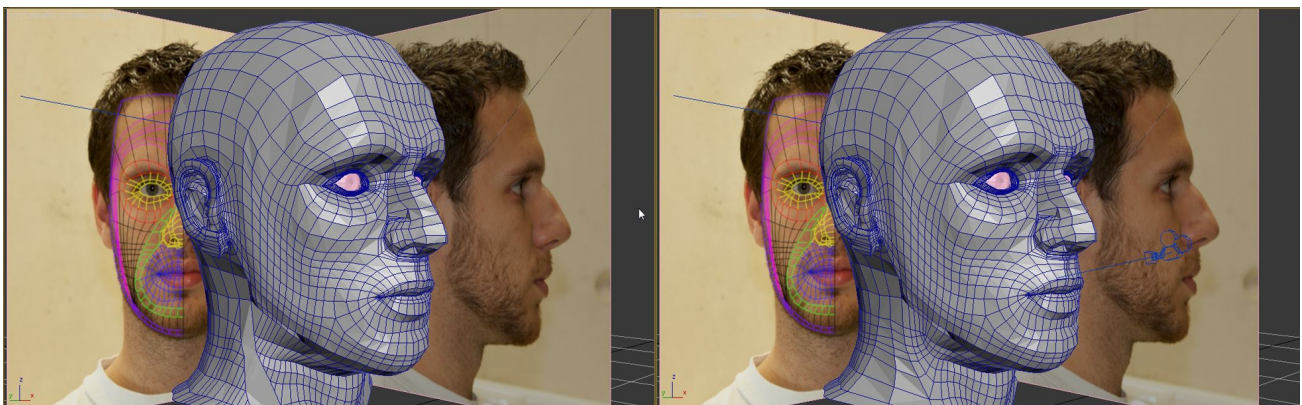
Seguidamente añadiremos más detalle a los pliegues de la parte central. Luego Construimos la parte exterior de la oreja, y realizamos lo que sería la parte de la piel que formaría parte de la cara. Se le da una forma aproximada al hueco que se ha formado en el cráneo.



Finalmente añadimos los últimos retoques para darle más realismo. Además controlamos la cantidad de edges que salen a la zona de la piel y ponemos la cantidad de edges que hay en el hueco asignado de la cara, para poder fusionarlo fácilmente. Las dos últimas imágenes son las que se producen al usar un modificador como turbosmooth para poder suavizar la malla final, como se hace en el caso de la subdivisión.

2.5.6 Modelado final

En esta parte ya toca añadir los últimos retoques a los modelos para que estén completos y sean correctos. Podemos elegir cualquiera de los dos modelos anteriores, ya que a partir de ahora siguen los mismos pasos de refinamiento. En este caso seguiremos desarrollando el último modelo, el creado por edge modeling.

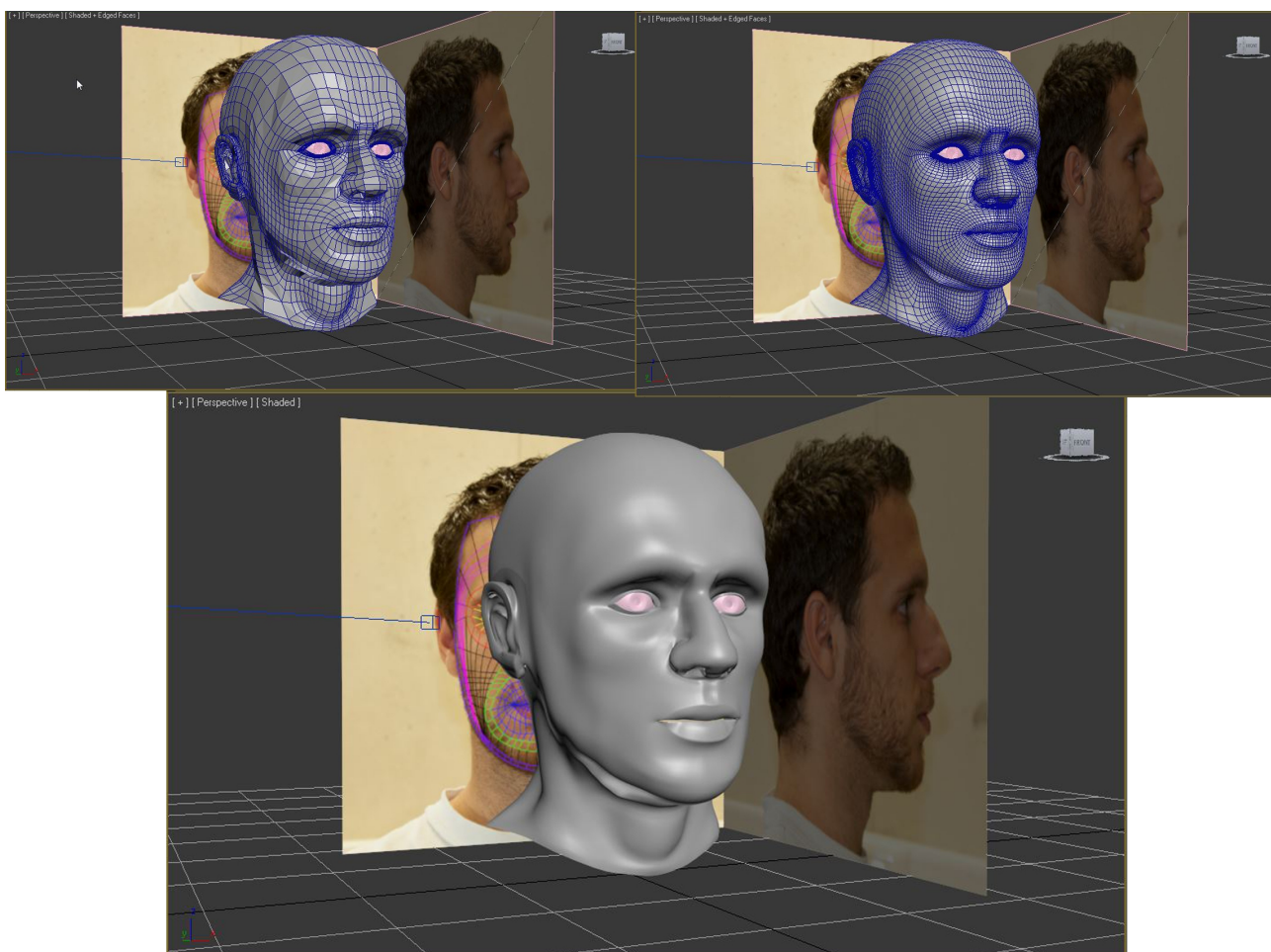


Lo primero que haremos será acoplar la oreja al modelo. Para ello lo incluiremos dentro del modelo y soldaremos los vértices ya que previamente hemos hecho el trabajo de que coincidiesen en número de edges compartidos. Tras ello nos propondremos realizar los dos últimos flows de la cabeza, que se encuentran en el cuello. Ellos nos darán la

movilidad necesaria para poder mostrar naturalidad al mover la cabeza y a que las formas de los músculos sean mas realistas. Además, y no por ello menos importante, nos permitirán reducir la cantidad de geometría que llevamos al tronco. El primero de ellos es el que forma la nuez. Todos los edgeloops que salen de la boca hacia abajo, y parte de los que salen de los ojos y que luego forman la parte de la papada los unimos para hacer un flow en U e irlos concentrando. Algo que hay que tener en cuenta al crear un modelo de un personaje en la zona de la papada, sea cartoon o realista o del estilo que sea, es que no debe ser completamente recto, sino que debe de hacer algo de curva, ya que si marcamos demasiado la mandíbula y no le damos una mínima curva a la unión entre la papada y el cuello, se verá extraño, robótico.

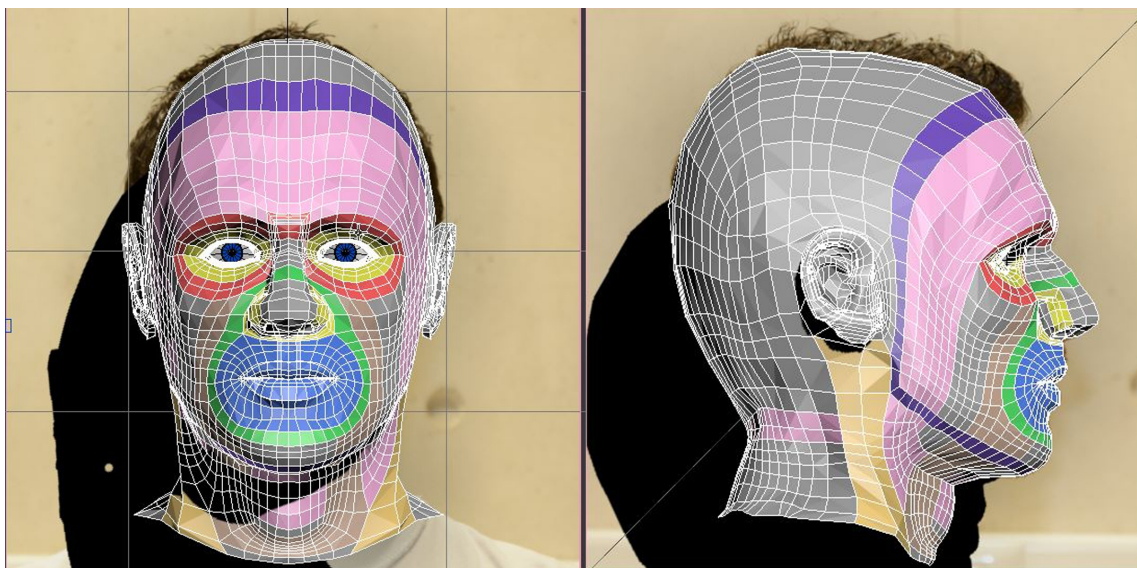
Además de la nuez, debemos crear un último flow para el esternocleidomastoideo, que atraviesa el cuello desde la zona de debajo de la oreja hasta la zona frontal. Al crear ese flow que atraviesa todo el cuello, una buena cantidad de loops se quedan sueltos y hace falta redirigirlos hacia algún lado ya que no pueden acabar ahí. Para ello volvemos a la solución usada anteriormente para crear el rostro, usaremos el esquema en X, que atravesarán hacia la parte de atrás del cuello para juntarse con su simétrica y así cerrarlos. Añadiremos algo más de geometría en el último flow creado, y resaltaremos y daremos los últimos retoques como realzar mandíbula, la nuez, el esternocleidomastoideo o la parte de la espalda.

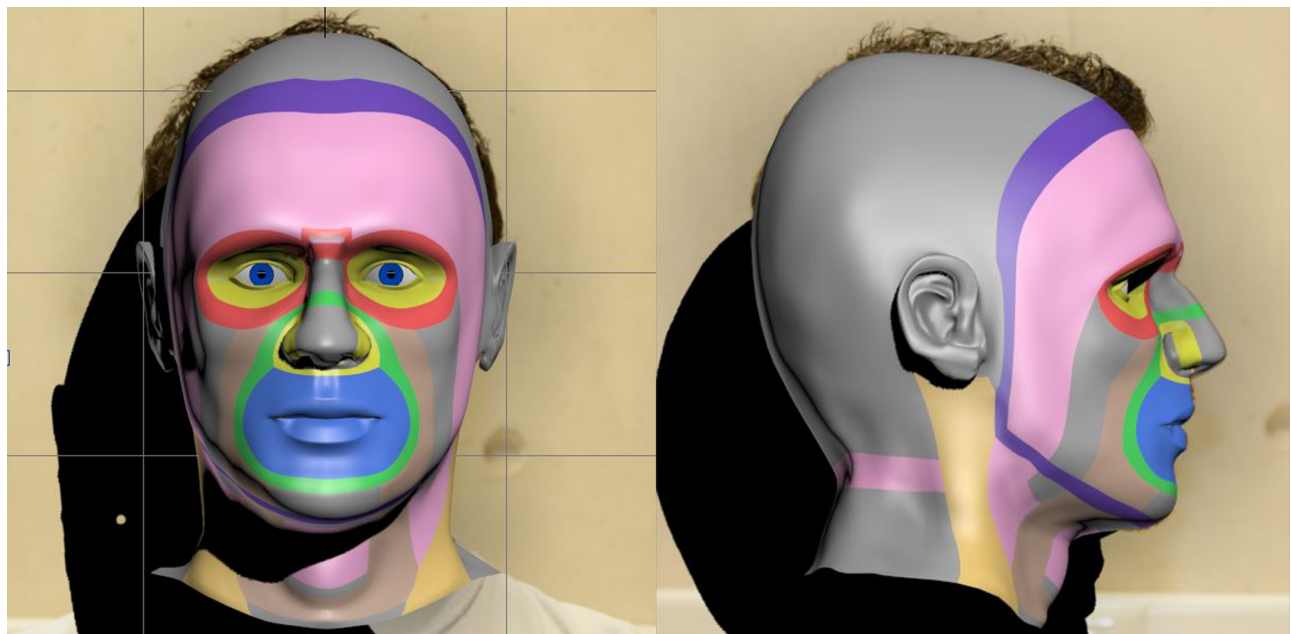
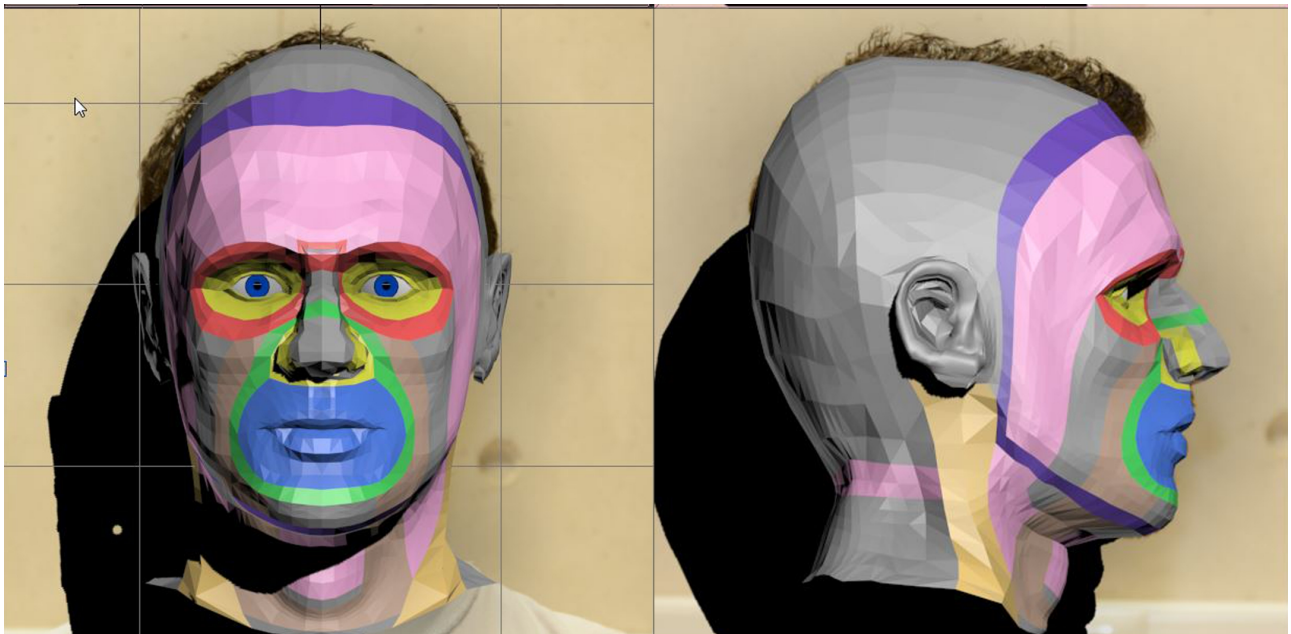
Para finalizar haremos el hueco de la boca, para ello simplemente tendremos que prolongar el borde de los labios hacia adentro y hacer una cavidad.



Finalmente al modelo se le puede aplicar un modificador como turbosmooth para crear un modelo de subdivisión. Con un sólo nivel de profundidad ese es el nivel de detalle que conseguimos.

Para finalizar, colorearemos los diferentes flows tal y como lo habíamos planeado en la etapa de premodelado, con los mismos colores, para que se vea el resultado final.







Topología, modelado y rigging facial

2.5.7 Conclusión Final

Las diferencias entre los dos métodos son claras:

- Box modeling
 - Ir desde lo grande hacia lo pequeño
 - Es más complicado seguir plantillas ya creadas
 - Lo importante es la forma
 - Es muy rápido
 - Muy recomendable para prototipado
 - Bueno para estudios de anatomía y de la forma
- Edge modeling
 - Empezar desde el detalle para ir creciendo
 - Lo más sencillo es empezar desde plantillas ya creadas, es mejor hacer un trabajo previo.
 - Lo importante es la topología
 - Tiende a ser bastante más lento que la otra forma
 - Muy bueno para detalles
 - Preciso

Al final las dos aproximaciones son correctas, depende de la situación en la que nos encontremos y el manejo del que lo modela para elegir una u otra. Es muy probable que para alcanzar la meta que necesitamos sea necesario combinar las dos formas (cosa que se ha demostrado por ejemplo si se hubiese seguido con el modelo de boxmodeling y se le añadiese la oreja, creada por edge modeling).

3. Rigging

3.1 Introducción: Morpher vs Bones

En esta fase del proyecto nos tenemos que centrar al rigging. El rigging es la técnica usada en animación 3d en la que un personaje esta representado por dos partes: una primera parte que es la representación de la superficie que es usada para dibujar al personaje (skin o la malla); y un sistema interno, normalmente conjunto jerárquico de huesos o bones (el conjunto se denomina rig o esqueleto), que se usa para ayudar a animar esa malla. Esta técnica suele usarse al animar humanos o modelos orgánicos, pero al ser un proceso orientado a hacer más intuitiva la creación de la animación, se usa para cualquier tipo de modelos en los que se quiera controlar la deformación de la malla.

En nuestro caso, la malla ya la hemos hecho, y sólo hará falta crear el sistema que nos permita animar.

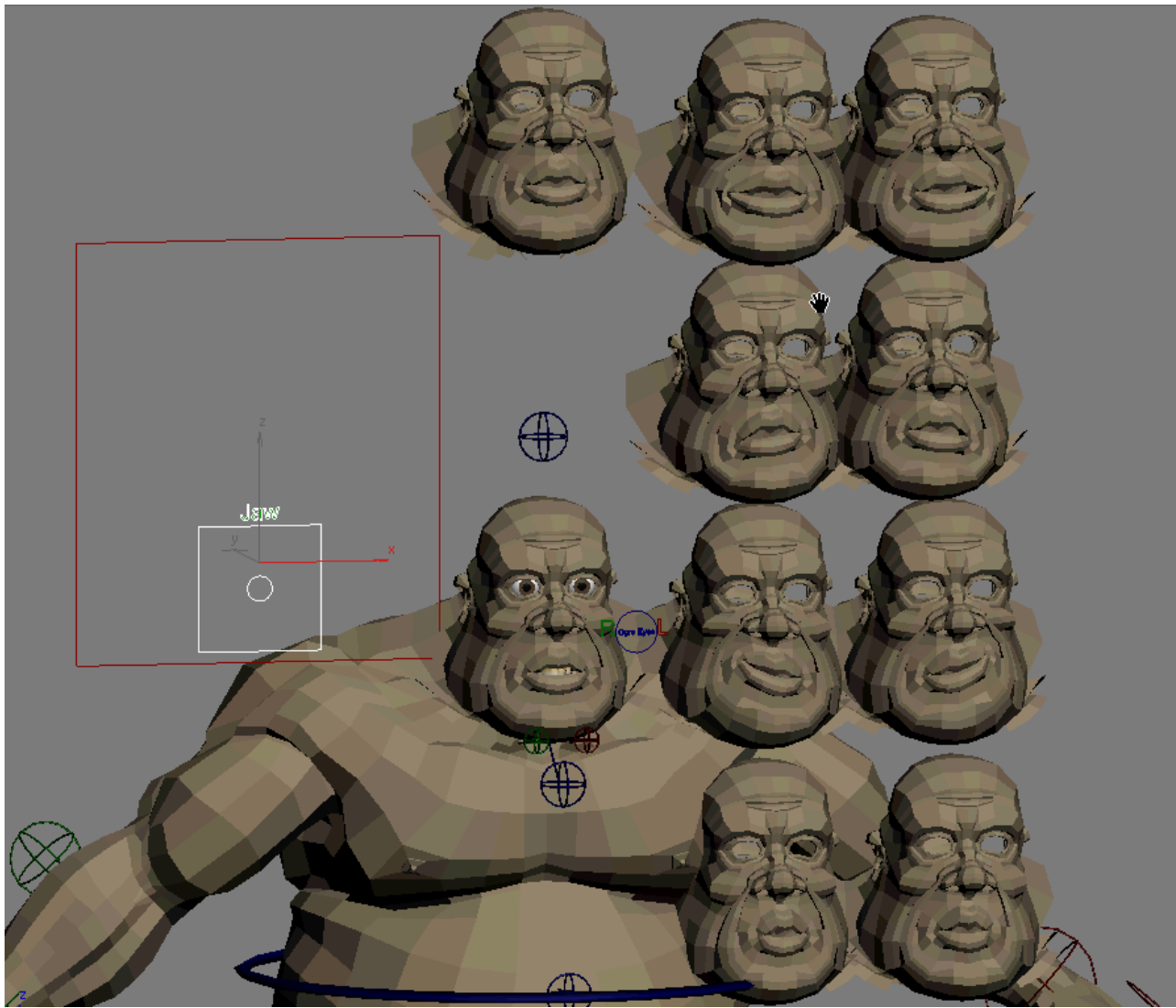
En el rigging facial, que es la parte que nos atañe a nosotros, hay dos posibles técnicas: basada en morpher, y basada en bones. De aquí elegiremos una y la desarrollaremos completamente según las ventajas que nos proporcione cada uno de ellos.

Empezamos con el morpher. El modelo que lo usa es una version deformada de la forma. Esto, aplicandolo a la cara, significaría crear una primera cara con una expresión neutra, para posteriormente ir creando nuevas caras con los diferentes gestos que busquemos. Para ello deben tener exactamente el mismo número de vértices, ya que cada vértice antiguo será asignado a uno del nuevo y al aplicar el morpher ocupará su misma posición y se fusionarán los dos modelos. Para cada una de las expresiones o nuevas formas hará falta crear un nuevo modelo, por lo que suelen hacer falta una gran cantidad de ellos para tener un rigging más o menos completo. Estas expresiones se consiguen moviendo, rotando o escalando los vértices generalmente, aunque también se consiguen a traves de movimientos de esqueleto.

Esta técnica tiene una serie de ventajas. El artista tiene más control sobre los movimientos porque define posiciones individuales de los vertices a traves de los diferentes keyframes. Esto puede ser útil al animar ropa, piel o expresiones faciales ya que puede llegar a ser complicado realizarlo correctamente a través e huesos. Otra ventaja es que una animación realizada por una aplicación de 3D en ocasiones tiene que ser transferida a otra, por ejemplo para renderizar. En ocasiones los diferentes softwares no

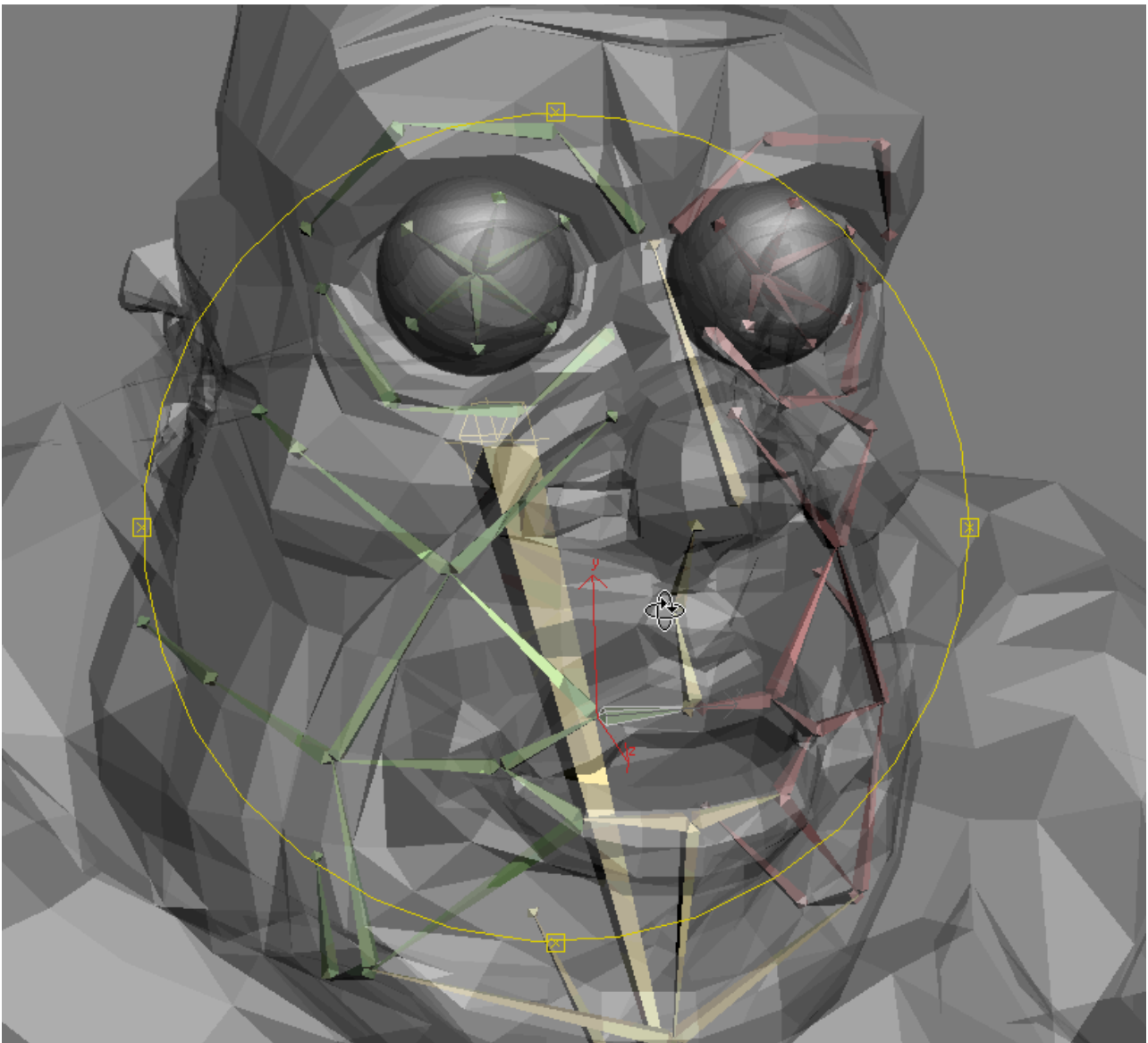
comparten estructuras internas como bones y otros efectos especiales, por lo que la transferencia a través de morph suele ayudar a evitar problemas mayores.

Las desventajas, en cambio, son también varias. La primera, y más determinante, es la cantidad de trabajo que supone. Requiere una labor mucho más intensa ya que debemos crear un modelo nuevo para cada uno de los gestos que queramos reflejar, y además las modificaciones de los modelos vértice a vértice son realmente muy costosas. Por ello hay dos variantes que se pueden combinar entre ellas, dependiendo a la meta que queramos llegar. Está la creación de poses absolutas, y la creación de poses relativas. La vertiente absoluta requiere crear un pose para cada uno de los gestos que busquemos, que es recomendada si va a ser usada en personajes que vayan a realizar pocos movimientos y van a tener poco protagonismo. Por el otro lado, podemos usar las poses relativas. Esta orientación sirve para realizar posturas clave que, combinadas entre ellas, dan paso al resto de posturas necesarias, por lo que con muchas menos consigues una gran cantidad de poses. Por último se pueden combinar las dos, usando las relativas para el funcionamiento general del personaje, y creando las poses absolutas de los gestos más característicos que necesitemos una gran precisión en ellos y no sea posible llegar a ellos a través de las relativas. Hay que tener en cuenta que hace falta también crear poses correctivas. A pesar de ello, por ejemplo para cerrar los párpados hacen falta un mínimo de doce modelos diferentes. El número de vértices debe ser el mismo, y estar numerados igual (por lo que no se pueden usar herramientas como mirror), ya que asignará cada vértice al del mismo nombre y las deformaciones de cada uno se verán reflejadas posteriormente. Los modificadores que los usan suelen estar limitados, así como el número de modelos previamente hechos. Con todo ello se puede decir que es complicado tener un sistema preparado completamente para producción, ya que se quedan con sistemas parcialmente completos. Por último cabe resaltar el hecho de que las fusiones entre las diferentes poses en una animación, en ocasiones crean distorsión, que no sucede con los modelos con huesos. Esto también se debe a que la velocidad que tiene el modificador es limitada por lo que no se pueden hacer cambios muy bruscos.



Por el otro lado tenemos la animación basada en huesos. Esta técnica consiste en crear una serie de “huesos” (bones), cada uno de ellos con una transformación tridimensional (que incluye su posición, escala y orientación), y un parent bone opcional. Así se pueden formar jerarquías. El producto de la transformación de un nodo hijo es la combinación de la combinación de su nodo padre y de la suya propia. Así que mover su nodo padre implicará mover el nodo hijo (ésto tiene una gran importancia como veremos posteriormente). Las jerarquías están compuestas por forward kinematics e inverse kinematics que interactúan entre ellas. Cada uno de los huesos está asociado con una porción de la representación visual del personaje. Este proceso de creación de la asociación se llama skinning. Cada bone tiene asociados un grupo de vértices, y éstos vértices pueden estar influenciados por varios huesos a la vez. El cálculo final de la posición de cada vértice viene dado por un cálculo de la posición de cada uno de los bones, y del peso que le es asignado.

En comparación con el morpher, es más complicado de realizar un skinning correcto que haga que los músculos se deformen tal y como se quiere. Además tiene mucho trabajo la fase de planificación de dónde poner cada hueso y por qué ponerlo ahí, ya que hay que hacerlo según los resultados que busquemos. A todo ello hay que sumarle el uso de las jerarquías, ya que eso condiciona bastante. Y el proceso de skinning es algo realmente delicado, ya que lo que en algunos movimientos nos viene mejor tener el peso sobre un hueso, en otros es mejor sobre otro, por ello es tan importante la parte de la planificación. Técnicamente es también mucho más complicado, porque es necesario el uso de bastantes más herramientas para que todo funcione correctamente. A pesar de todo, una vez hecha esa planificación, la cantidad de trabajo es bastante menor ya que no hace falta crear tantos modelos y podemos conseguir unos resultados profesionales.

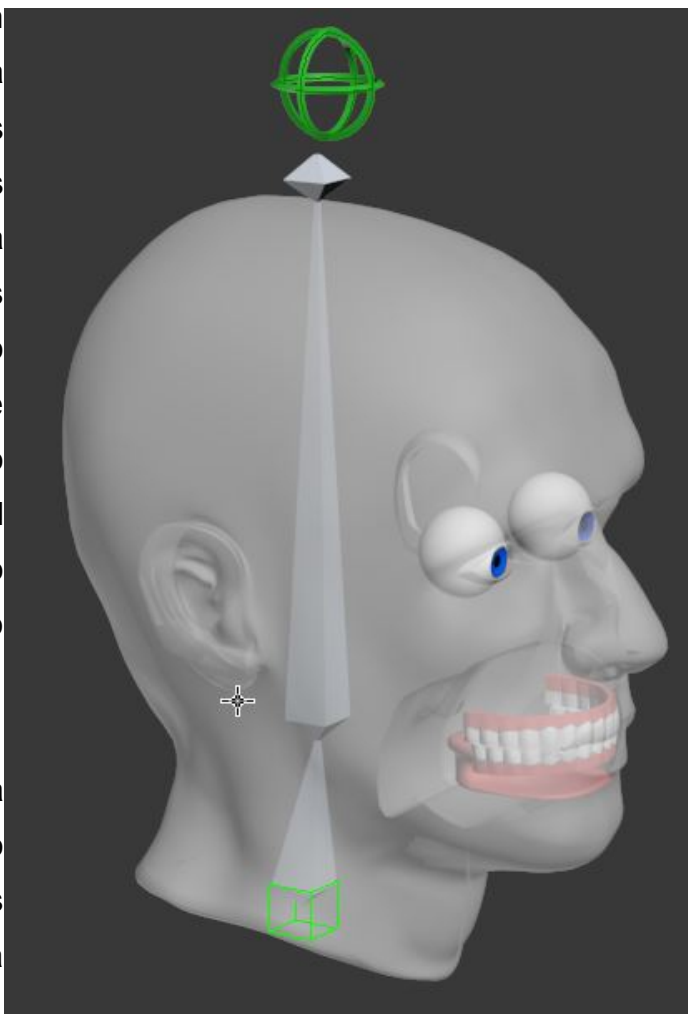


Tras sopesar las dos opciones, escogemos la segunda, ya que la relación de tiempo invertido y resultados finales es la mayor, aunque sea el reto más técnico de los dos. El hecho de tener que hacer tantas poses para cada uno de los movimientos es el factor clave, ya que no se dispone de tanto tiempo. Además, con el sistema de huesos y el interfaz que vamos a crear, en un futuro se puede ampliar gracias al sistema de representación a la animación facial por captura de movimiento, poniendo los marcadores en los mismos lugares que corresponden e interactuando.

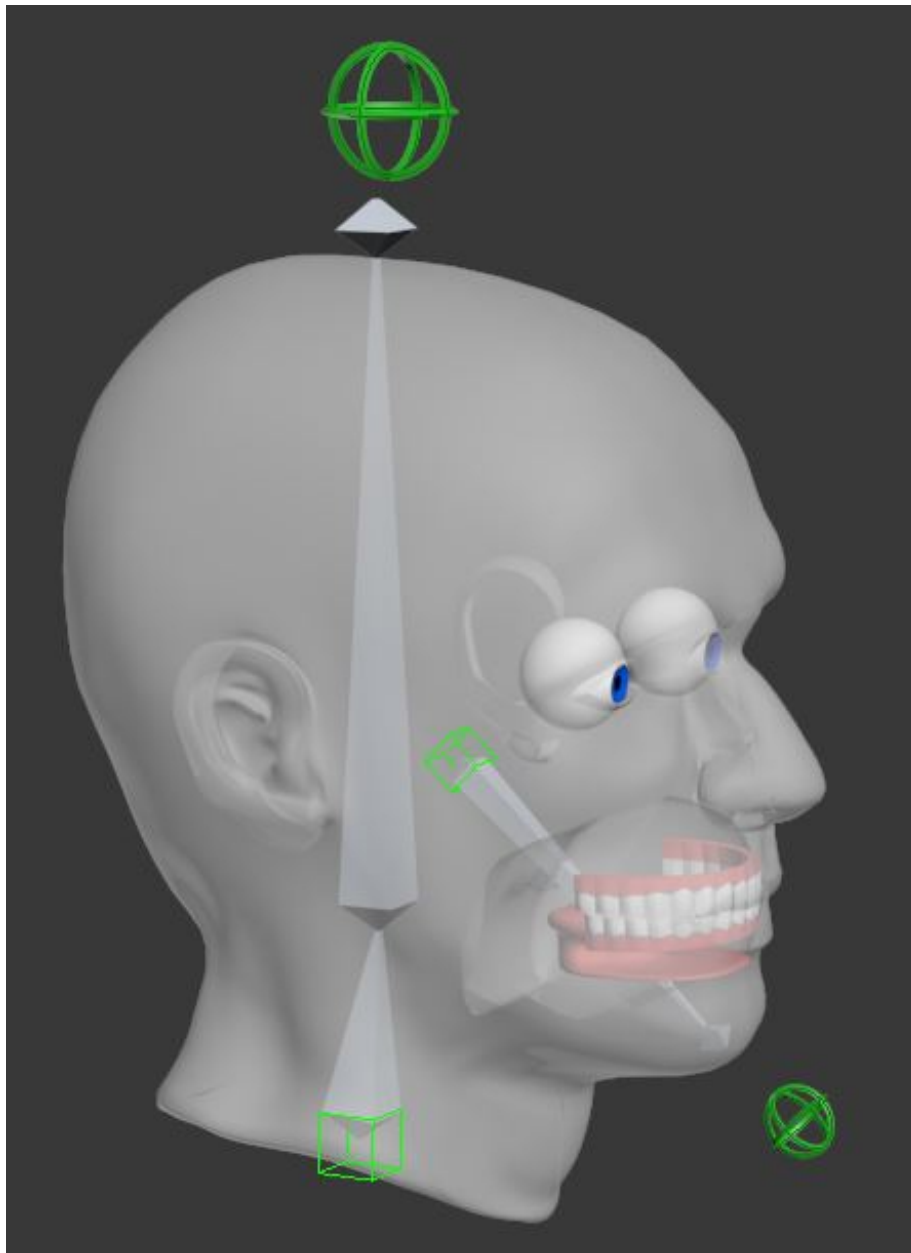
3.2 Rigging de la cabeza

El primer paso del proceso es crear los huesos principales. Esto es un paso común en los dos métodos, y nos proporcionará la movilidad del cuello y de la cabeza. Por ello lo realizamos en la zona central del modelo dos huesos, uno que nos servirá para el cuello, y otro para la cabeza, de abajo hacia arriba. Tendrán siempre una misma posición relativa ya que situaremos un point en la base, que estará siempre fijo. Ese punto en un modelo completo puede significar la posición del resto de huesos del cuerpo, por eso tendremos una serie de puntos que se mantendrán fijos representando lo que podría ser la unión con el resto del cuerpo. Los huesos se manejarán con un control que hemos creado con splines, y sólo rotarán. Para ello les aplicaremos a cada uno de los huesos un orientation constraint. Al hueso de la cabeza le asignaremos un peso de 100, mientras que el cuello en cambio tendrá un peso de 66 respecto al controlador y de 33 con el punto. Esto permitirá que el cuello rote menos que lo que lo hace la cabeza.

El siguiente punto clave de nuestra jerarquía es la mandíbula. Para ello crearemos un hueso que empezará más o menos desde la altura de la oreja hasta



la barbilla, para que al girarlo tenga el mismo movimiento que la mandíbula de verdad. A su vez le alineamos un punto al hueso raiz y linkamos ese hueso al punto, para que dependa en todo momento de él. Por último creamos un nuevo controlador como el anterior que servirá para manejar la rotación. La rotación del hueso se verá afectada por un orientation constraint del controlador, y para que el controlador se mueva a la vez que la mandíbula en este caso pondremos el pivote alineado en el punto.



También linkaremos la mandíbula superior al punto, mientras que la mandíbula y la lengua la linkaremos al hueso de la mandíbula. Lo siguiente que tenemos que hacer es controlar los ojos. Primero, con splines, crearemos tres controladores. Uno global, y dos

individuales. Los individuales indicarán a dónde se orientará cada ojo, mientras que el global servirá para mover ambos controladores individuales a la vez. A la vez, lo aprovecharemos para regular el tamaño de las pupilas. Para ello, lo hacemos gracias a MaxScript, el lenguaje de 3d Max. Creamos dos scripts para poder controlarlos.

```
def=attributes ModelEyes
(
    parameters eyesP rollout:eyesR
    (
        pupil type:#float UI:pupilSp
    )
    rollout eyesR "Eyes"
    (
        spinner pupilSp "Pupil"
    )
)

custAttributes.add $.modifiers[1] def

pupil_v2: Bloc de notas
Archivo Edición Formato Ver Ayuda

attributes ModelEyes
redefine: def
(
    parameters eyesP rollout:eyesR
    (
        pupil type:#float UI:(pupilSp,pupilSl)
    )
    rollout eyesR "Eyes"
    (
        local Rrange=[-80,80,0]
        spinner pupilSp "Pupil" range:Rrange
        slider pupilSl "" range:Rrange
    )
)

custAttributes.add $.modifiers[1] def
def=custAttributes.getDef $.modifiers[1].ModelEyes
```

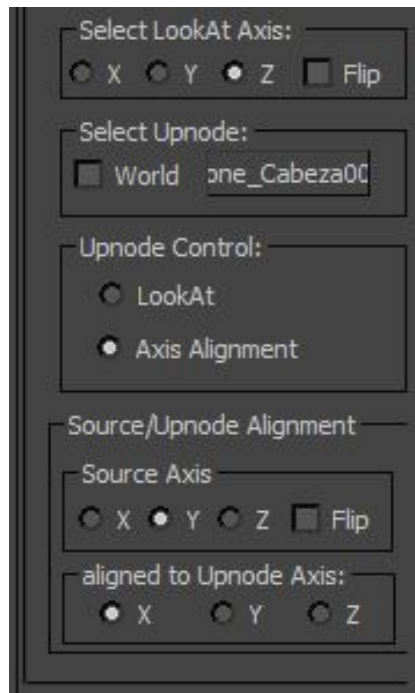
Con ellos podremos controlar las pupilas y limitarlas a un tamaño determinado para que ni nos quedemos sin pupilas, ni sean más grandes que el propio iris. El controlador que se queda así, con posibilidad de hacerlo a través de un spinner o de un slider.

Lo siguiente que hay que hacer para configurar los ojos es crear dos puntos, uno en cada ojo, que estarán linkados al punto

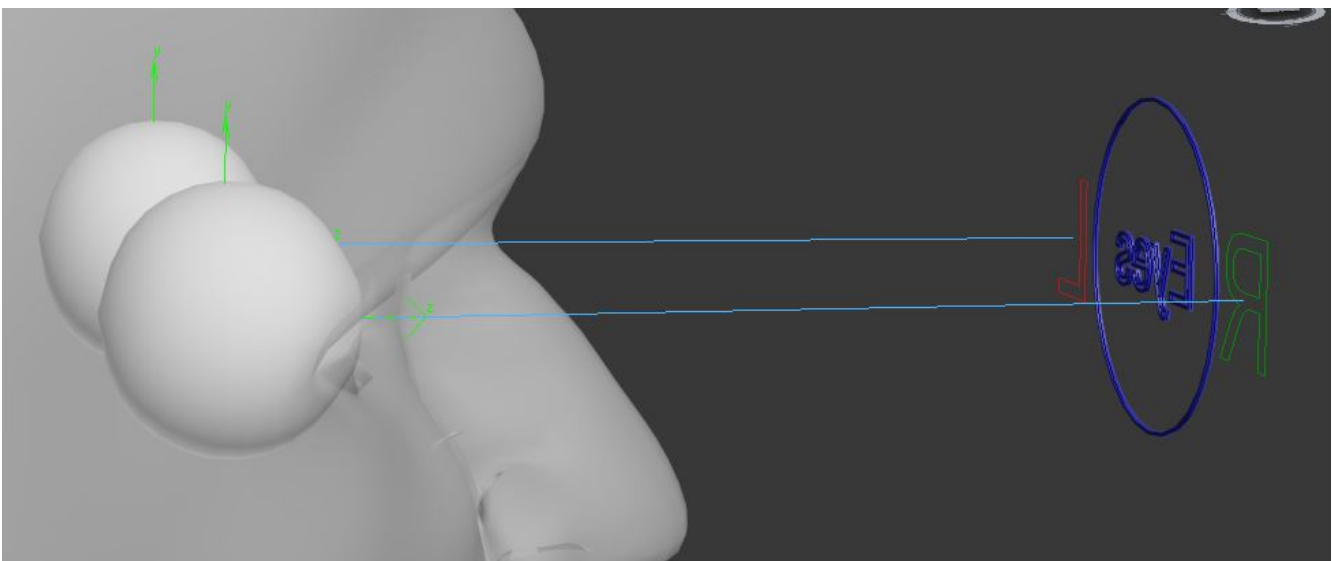


central donde se encuentra el inicio de la mandíbula.

Para que siempre estén mirando a los controles, les añadimos a cada uno un modificador de lookAtConstraint, siendo su Upnode el punto de la mandíbula (en este modificador se debe siempre poner como Upnode el elemento que esté por encima en la jerarquía que se esté creando, por eso por defecto está el mundo, pero más adelante este mismo concepto lo volveremos a usar).

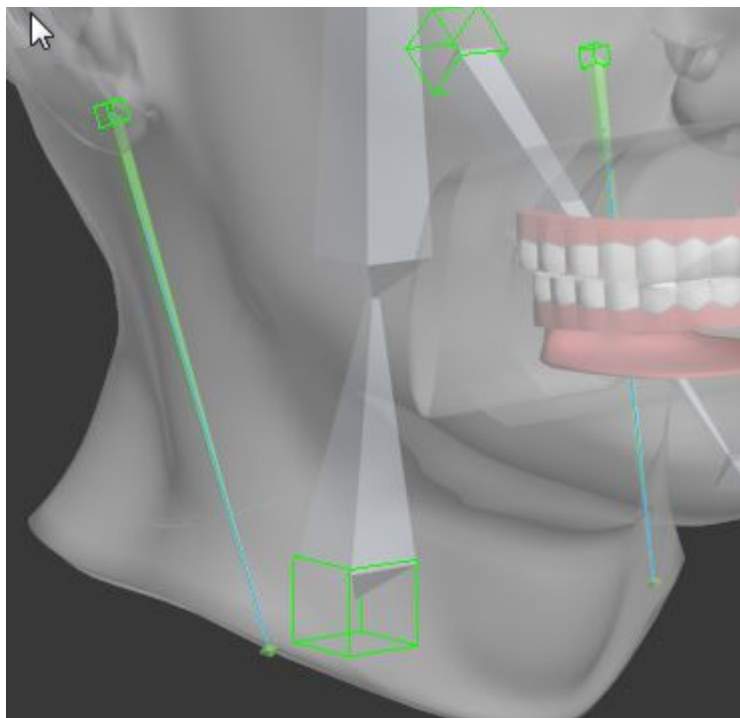


A su vez alinearemos los ejes para que no haya problemas en la animación posterior y miren hacia los objetivos, no en otra dirección. Con todo ello nos quedan los controles como en la imagen. El viewline lenght lo hemos reducido hasta los controladores, pero no los hemos hecho desaparecer para que queden claro la dirección de la mirada.

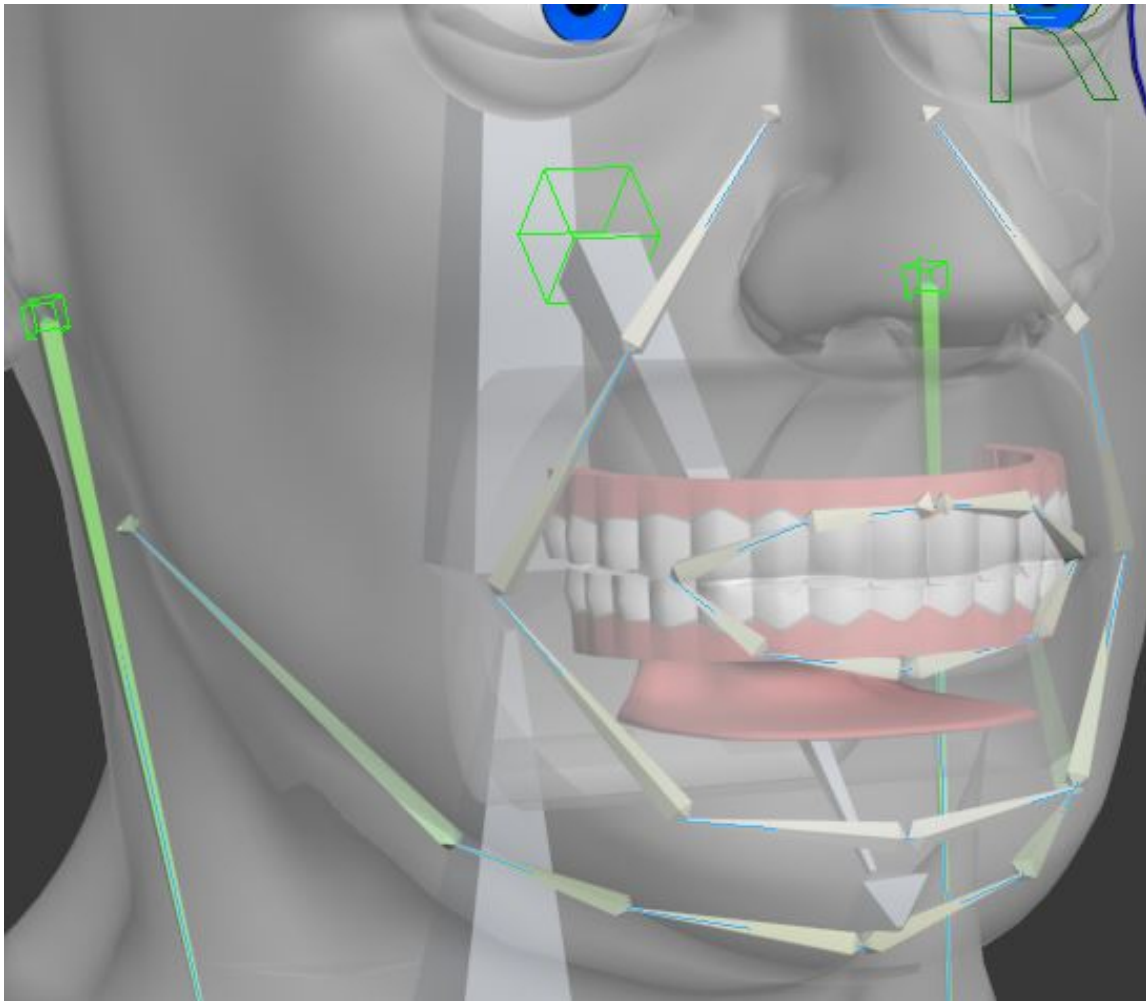


Hasta aquí el proceso es común para el método morpher y el de los bones, pero a partir del siguiente paso se verá la gran diferencia entre ellos. En el método bones, al haber realizado un modelado pensado en la animación, nos basamos en los edgeflows diferentes que pensamos que se podían animar. Lo único que debemos hacer será seguir esos edgeflows ya que nos permitirán conseguir una deformación correcta de la malla.

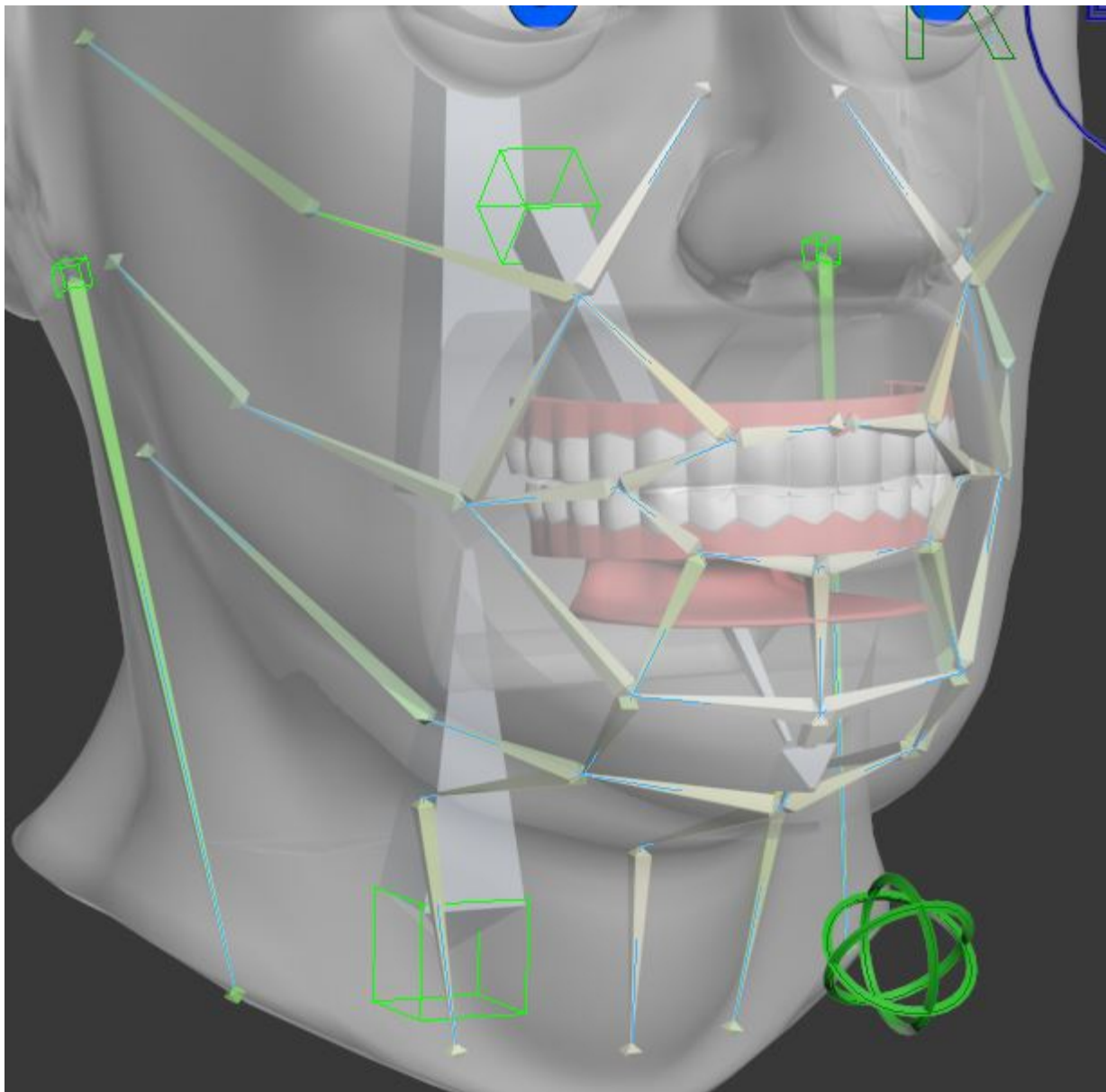
Comenzaremos con los más sencillos, ya que son los más independientes de todos ellos. Los primeros que vamos a realizar serán los del cuello, los que corresponden al esternocleidomastoideo, que los realizaremos de arriba hacia abajo siguiendo el curso del músculo. Todos los huesos los tendremos que configurar en el bone tool como squash para que se deformen como queremos, además de quitarles la casilla de frozen length. Haremos todas las creaciones sobre un lado, y las duplicaremos con la herramienta mirror tool del propio bone tool, ya que sino las coordenadas se modifican inversamente. Para modificar los tamaños del hueso habrá que usar el bone edit, y finalizado todo el proceso, cuando esté correctamente puesto, tendremos que resetear tanto el squash como el scale.



Los siguientes pasos serán crear los huesos que siguen los edgeflows de la boca, el nasolabial y la mandíbula. Este ultimo nos servirá para remarcar el contorno y que sea más facil el poder mover toda la parte del cuello al estirares y contraerse sin que pierda definición. Los otros dos nos proporcionaran suficiente funcionalidad para poder mover la boca y provocar una sonrisa, así como poder abrir y cerrar la boca y moverla hacia los lados.

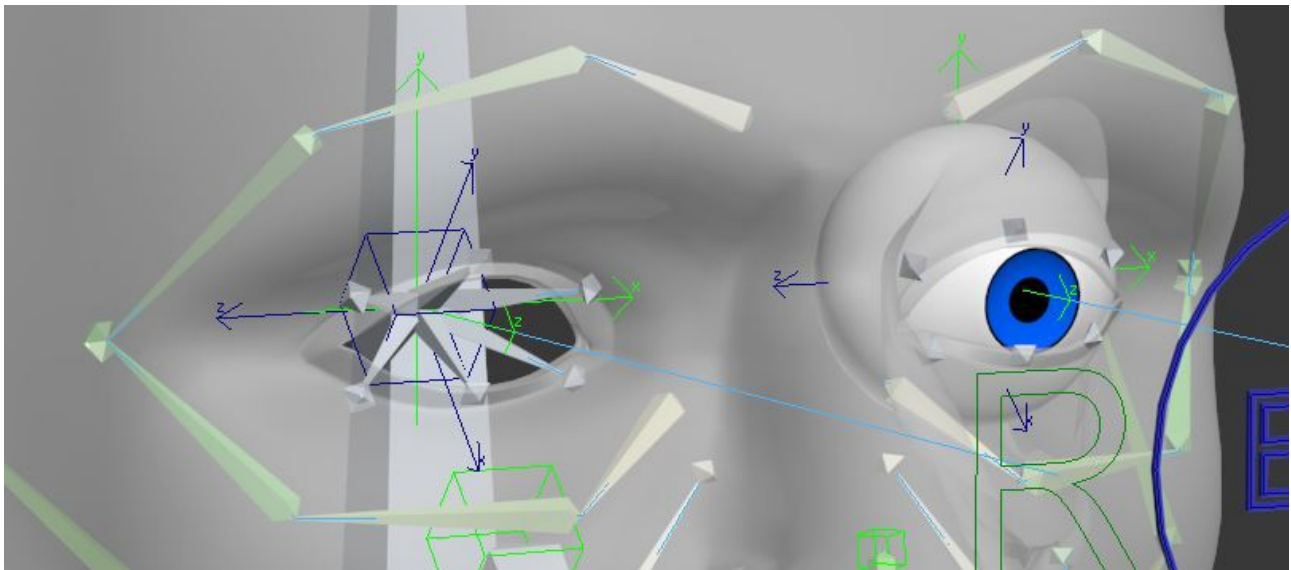


El siguiente reto es crear las uniones entre los diferentes loops, así como crear huesos suficientes para poder riggear correctamente el cuello y poder añadir la funcionalidad de la garganta. Además también debemos añadir músculos que permitan controlar la parte de la mejilla y los pómulos. Por último añadimos un músculo para la parte del bigote. Con ello nos queda la siguiente estructura de músculos.



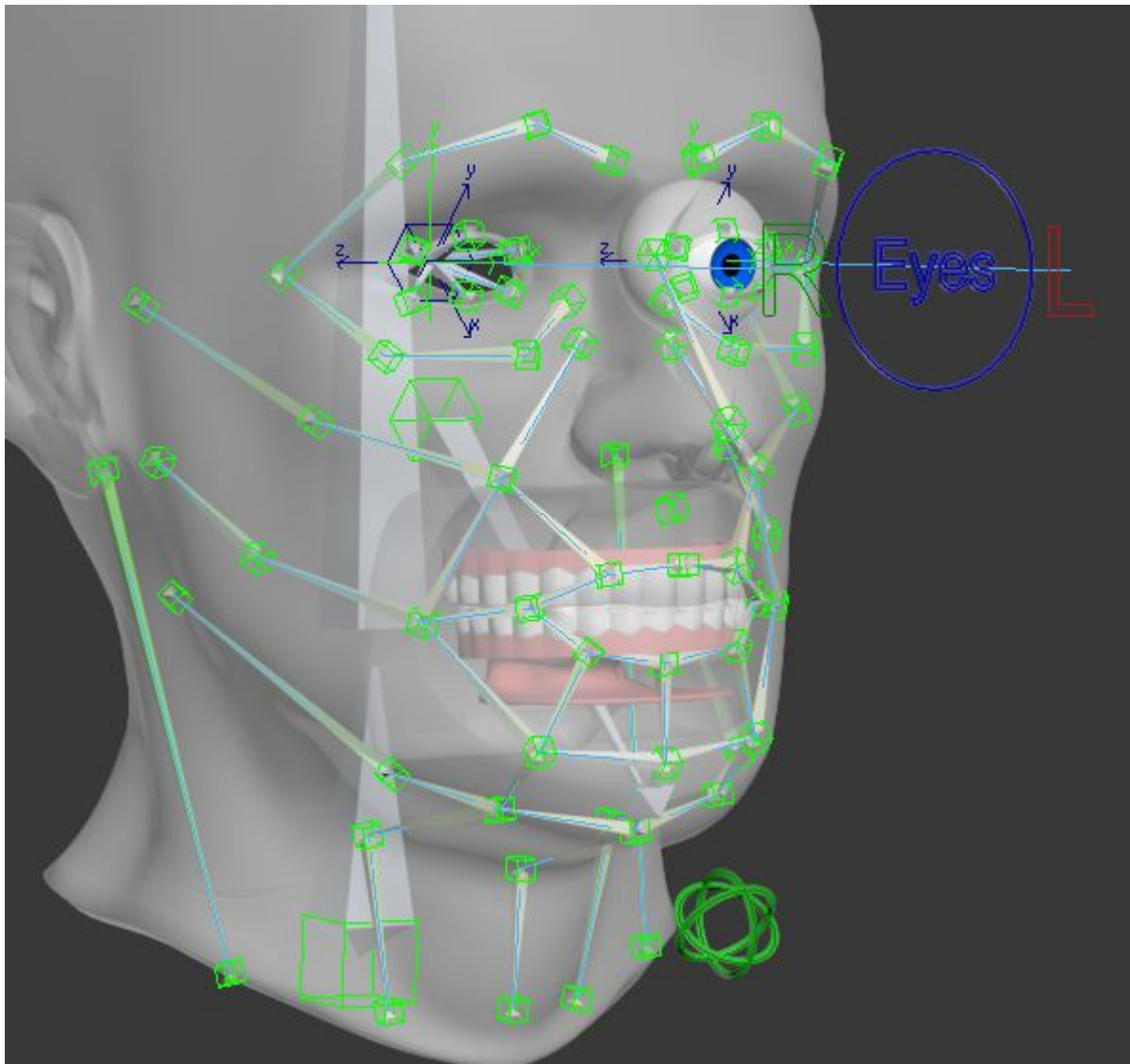
La última zona que hace falta poner huesos es la parte de los ojos, así como la nariz. El de la nariz es un hueso simple, como eran los del esternocleidomastoideo. Luego están los huesos de las cejas y las órbitas, todo ello sin dejar de seguir los diferentes flows. Para que el movimiento de los párpados sea realista, la mejor es crear los huesos para que roten respecto a un punto. Estos huesos deben ser siempre de longitud fija. Lo que debemos hacer es crear dos nuevos puntos, dependientes del punto de la mandíbula, que además de convertir sus coordenadas a cero, les tenemos que añadir un orientation constraint para que roten a la par que su predecesor, y así los párpados seguirán en el mismo plano que la cabeza. Para cada uno de los párpados le haremos tres jerarquías de huesos diferentes, dos laterales y una central, todas con origen en el nuevo punto creado,

por lo que rotarán todas a la vez y sobre ellas mismas.



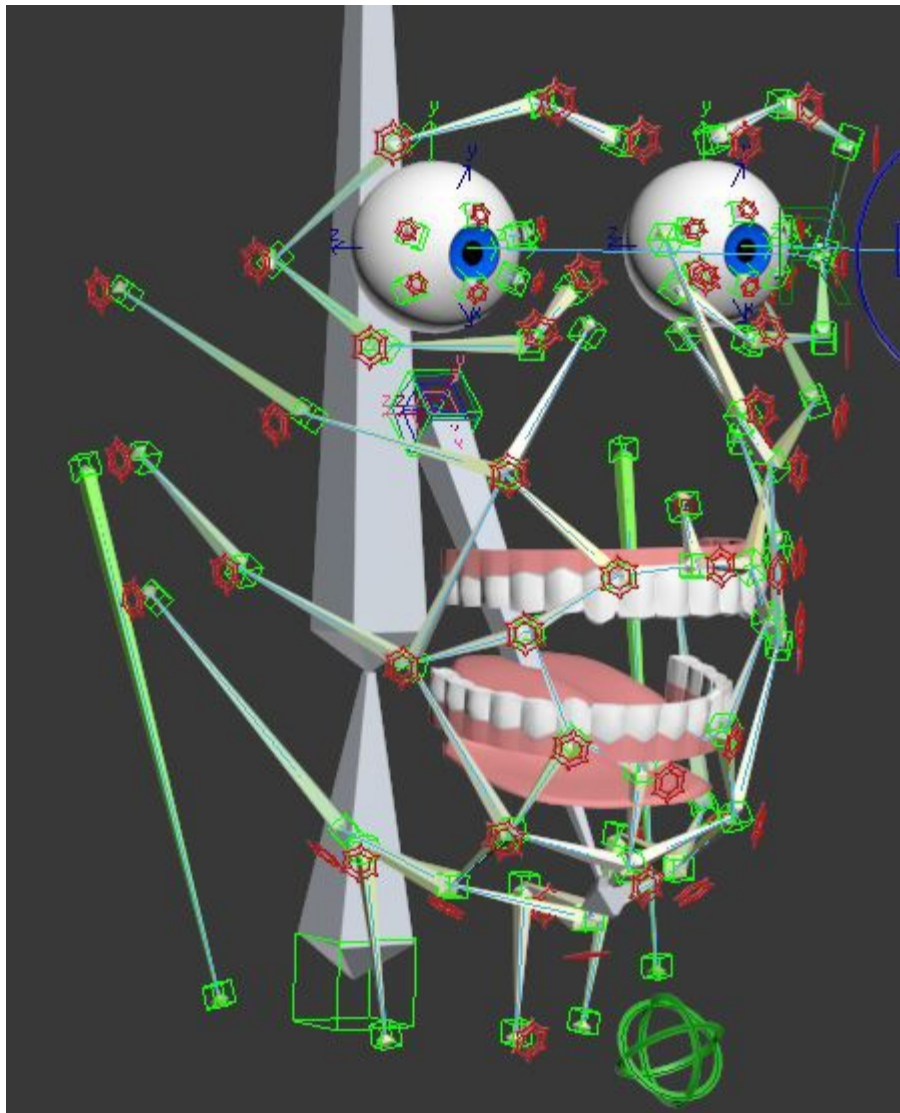
Con esto tendremos todos los huesos creados. Toca renombrarlos para que podamos acceder fácilmente a ellos y no liarnos al usarlos, ya que son una gran cantidad de ellos. Ahora tenemos que preparar el terreno para crear el interfaz para poder mover correctamente los huesos. En el estado en el que está ahora mismo, si movemos un hueso que sea raíz, o lo rotamos, rotaremos o moveremos toda la jerarquía, mientras que si movemos otro que sea hijo, moveremos toda la jerarquía que dependa de él pero el hueso anterior no apuntará hacia él, por lo que habrá una deformación que no nos interesa. Además hay puntos que están “compartidos” por varias jerarquías diferentes (por ejemplo en el caso de la boca y mentón). El efecto que buscamos es que al mover determinado controlador, se muevan los huesos que dependan de ese controlador, y el resto se queden fijos. Para ello tenemos que añadir un punto al principio de cada uno de los huesos, o por lo menos de las uniones de varios. Todo esto lo tendremos que hacer con el bone edit, para poder modificar el tamaño de los huesos sin mover el resto. Crearemos un punto, y lo alinearemos a cada hueso, si tiene más huesos en común, los alinearemos a ese punto creado. Tras ello les añadiremos un position constraint a cada hueso. Por último añadiremos un lookat constraint a los huesos padres de la jerarquía, teniendo como upnode su propio punto, y mirando hacia el nuevo punto creado. Así nos aseguramos que en todo momento, al mover el punto, se modifican todos los huesos que le afectan, pero sin romper jerarquías ni hacer deformaciones extrañas. Este proceso lo debemos hacer con cada hueso, como hemos dicho anteriormente. Los puntos los

deberemos tener rotados en perpendicular a lo que sería la superficie de la cara. El resultado es el siguiente:



Tras esto debemos añadir los que serán los controladores de esos puntos, lo que el animador podrá ver para poder mover y modificar toda la disposición, ya que el resto estará oculto. Para ello creamos un nuevo controlador a base de splines y le damos un color característico, que se pueda ver bien. Los iremos alineando y colocando en todos los puntos que necesitamos mover algo manualmente (ejemplo, en el esternocleidomastoideo no ya que el movimiento lo haremos automático, o en la parte superior de la nariz tampoco, ya que esa zona no tiene movilidad alguna, deberemos ponerlos en los sitios en los que deseamos tener una movilidad especial). La gran mayoría de ellos se podrán mover y trasladar normalmente, pero los de los párpados se deberán rotar para poder conseguir que funcionen correctamente. Lo siguiente que debemos hacer es linkar esos

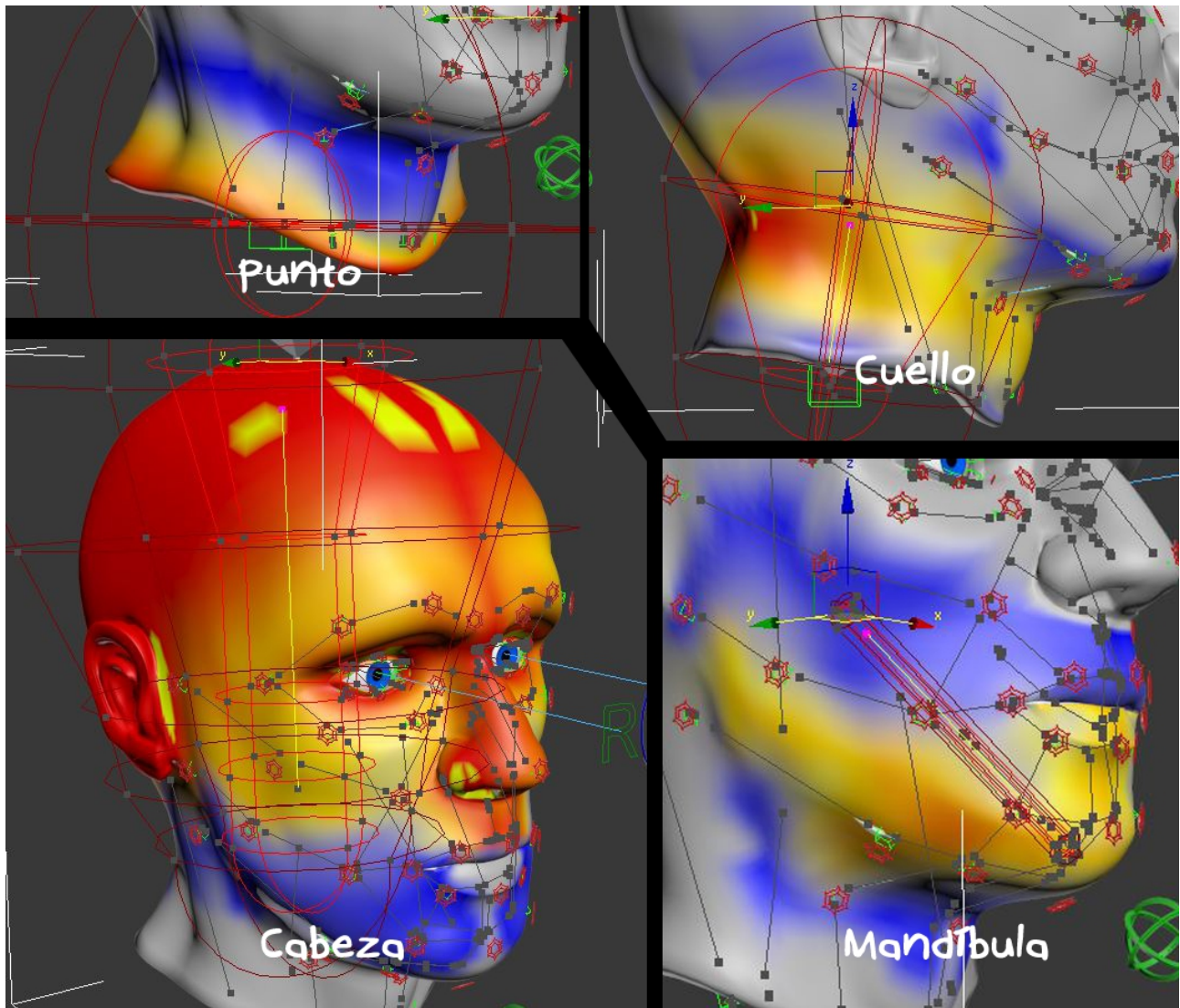
puntos creados a los sitios que queramos que dependan. Por ejemplo, los puntos de la parte inferior del cuello (exceptuando la nuez) los linkaremos al punto fijo de la base del cuello. Los puntos superiores a esos se linkarán al hueso del cuello. Los huesos que definen el contorno de la mandíbula se linkarán al hueso de la mandíbula. Crearemos tres puntos más, todos alineados al punto del inicio de la mandíbula, con un position constraint a ese mismo punto, y un orientation constraint de ese punto y de la mandíbula, con diferentes grados cada uno. Eso hará que al mover la mandíbula, esos puntos se muevan progresivamente. La explicación a esto es que al mover la mandíbula, no todos bajan al 100%, sino que hay algunos que se quedan a medio camino, o que se mueven levemente. El resto lo linkaremos a la cabeza. Los de los párpados los linkaremos a los últimos puntos creados en los ojos. A su vez pondremos el pivot point de los indicadores en el punto del ojo, para que roten sobre él.

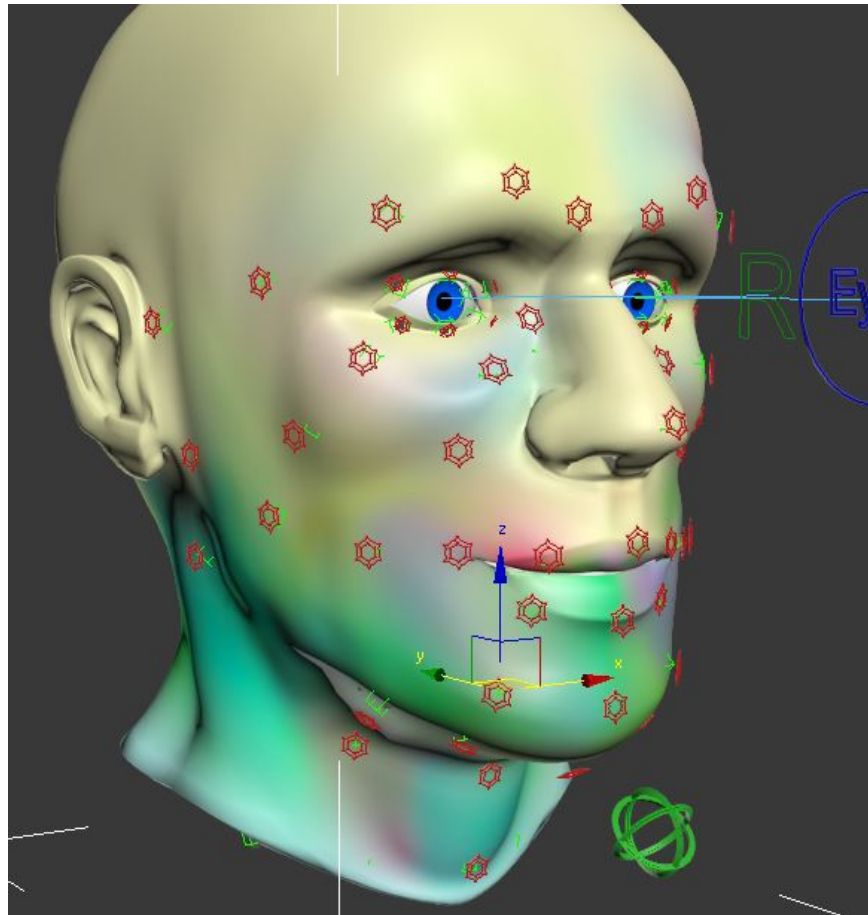


Con todo esto creado, ya tenemos la estructura interna de todo el sistema. Ahora

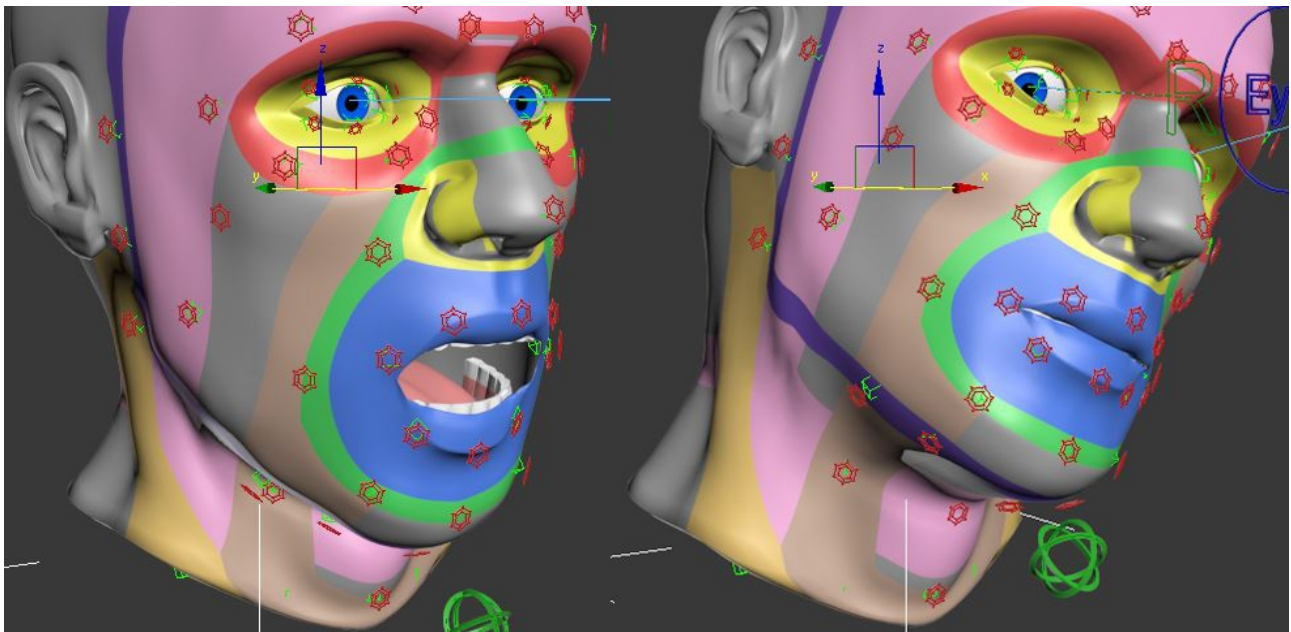
debemos asignar el último paso, pero uno de los más costosos y lentos. El último paso es aplicar el modificador skin en la cara. El primer paso que tenemos que hacer es añadir todos los huesos al modificador skin. Además tendremos que añadir el punto del hueso del cuello. Lo añadiremos ya que queremos tener la parte inferior del modelo fija. Para poder sacar un buen resultado en el modificador skin, lo mejor es ir parte por parte, realizando movimientos precisos de los límites que estamos buscando en las diferentes articulaciones. Eso lo debemos hacer progresivamente y con diferentes movimientos específicos, además de usando cada una de las herramientas posibles. La primera que debemos usar es los envelopes, con el que conseguimos una disposición inicial de los pesos de los huesos en la malla. Los valores de los pesos varían entre cero y uno (representados por un gradiente de colores entre azul, amarillo y rojo), siendo cero un hueso que no le afecta nada, y uno un hueso que le afecta totalmente a ese vértice. Una vez que tenemos una idea inicial de como queremos todo, pasamos a bakear los vertices, eso hace que fijemos el valor actual de esos vértices y ya no dependamos más de los envelopes y podamos usar el resto de herramientas. Las dos herramientas que podemos usar a partir de ahora será el uso de valores absolutos de los vértices y el paint weight. Con la primera asignaremos individualmente los valores a los vértices, mientras que con la otra iremos “pintando” los valores con aumentos o disminuciones progresivas de los valores, con el mismo concepto que podemos encontrar en un softselection. Con ello iremos buscando un ajuste total de los vértices para que se deformen correctamente hasta en las posiciones más extremas sin haber anomalías, ya que se pueden colapsar entre ellos y superponer diferentes caras en los pliegues, o estiramientos incorrectos. Todos los cambios los aplicaremos sobre un lado de la cara y con la herramienta mirror del propio skin traspasaremos los valores a los vértices correspondientes del otro lado.

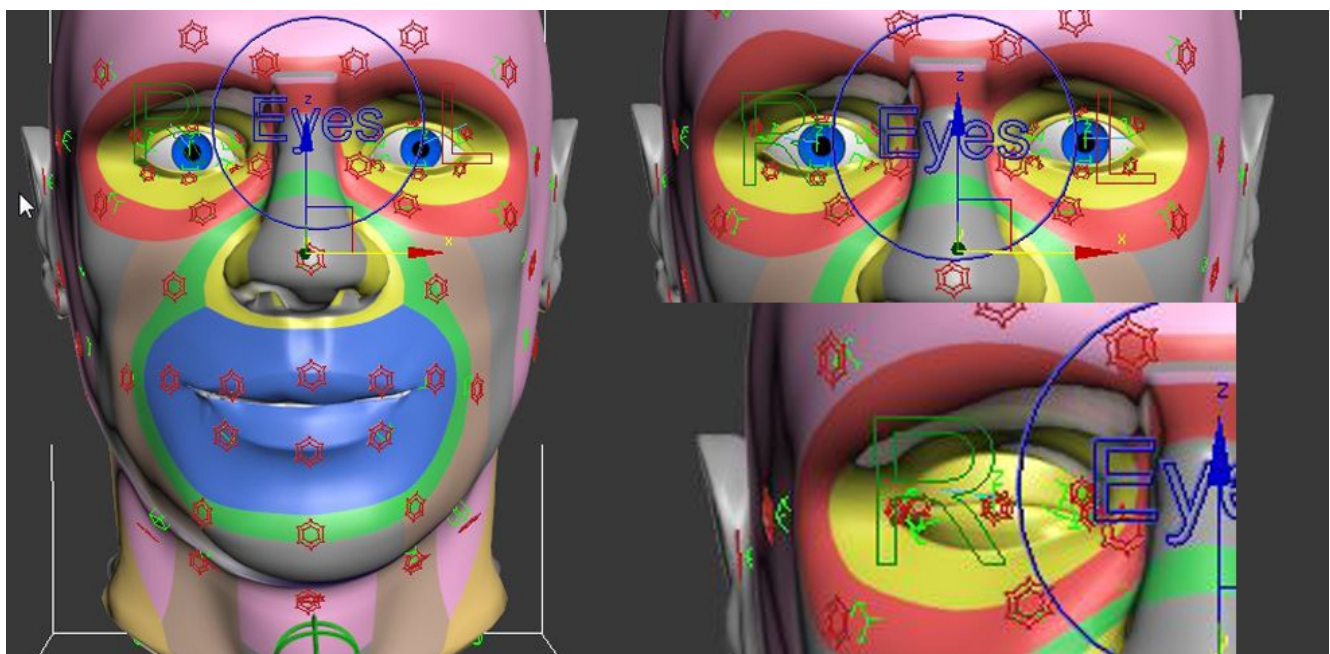
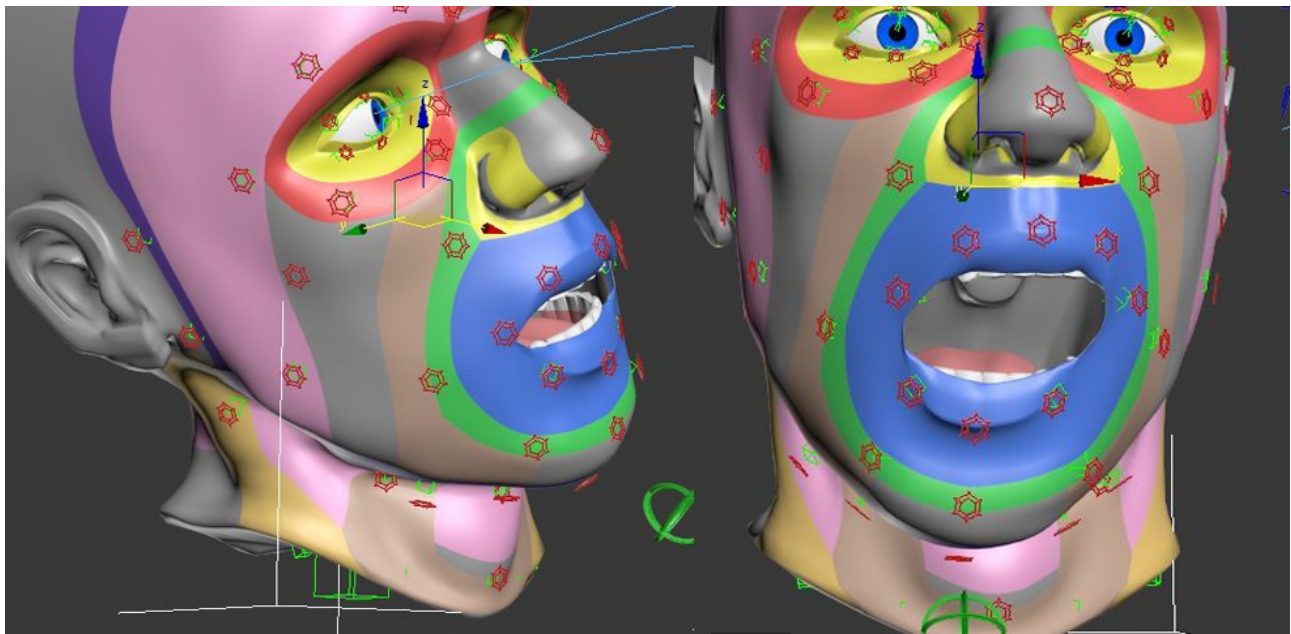
Estos son los cuatro huesos principales y sus pesos finales asignados, y en la segunda imagen tenemos un sistema en el que a cada uno de los huesos se les asigna un color y cada zona se ve reflejado cada una de las zonas de influencia.





Aquí podemos ver una serie de gestos que podemos llegar a conseguir.





4. Bibliografía

Conceptos generales de 3d

http://en.wikipedia.org/wiki/3D_computer_graphics

http://en.wikipedia.org/wiki/History_of_computer_animation

<http://lucky13production.blogspot.com.es/2009/12/production-workflow.html>

Diferentes tipos de estructuras y técnicas de modelado sobre ellos

http://en.wikipedia.org/wiki/Subdivision_surface

[http://download.autodesk.com/us/maya/2010help/index.html?](http://download.autodesk.com/us/maya/2010help/index.html?url=Polygons_overview_Introduction_to_polygons.htm,topicNumber=d0e200792)

[url=Polygons_overview_Introduction_to_polygons.htm,topicNumber=d0e200792](http://download.autodesk.com/us/maya/2010help/index.html?url=Polygons_overview_Introduction_to_polygons.htm,topicNumber=d0e200792)

[http://download.autodesk.com/us/maya/2010help/index.html?](http://download.autodesk.com/us/maya/2010help/index.html?url=NURBS_overview_What_are_NURBS_.htm,topicNumber=d0e227690)

[url=NURBS_overview_What_are_NURBS_.htm,topicNumber=d0e227690](http://download.autodesk.com/us/maya/2010help/index.html?url=NURBS_overview_What_are_NURBS_.htm,topicNumber=d0e227690)

[http://download.autodesk.com/us/maya/2010help/index.html?](http://download.autodesk.com/us/maya/2010help/index.html?url=Subdivision_surfaces_overview_What_are_subdivision_surfaces_.htm,topicNumber=d0e247068)

[url=Subdivision_surfaces_overview_What_are_subdivision_surfaces_.htm,topicNumber=d0e247068](http://download.autodesk.com/us/maya/2010help/index.html?url=Subdivision_surfaces_overview_What_are_subdivision_surfaces_.htm,topicNumber=d0e247068)

http://es.wikipedia.org/wiki/Cel_shading

<http://theorangeduck.com/page/subdivision-modelling>

<http://es.wikipedia.org/wiki/NURBS>

<http://advmaya8tutorials.blogspot.com.es/2007/09/nurbs-v-polygons-v-subdivisons.html>

<http://www.digitaltutors.com/forum/showthread.php?19078-What-is-the-difference-between-NURBS-subdivisions-and-polygons>

<http://blendernewbies.blogspot.com.es/2008/11/3d-mesh-topology-tip-quads-vs-triangles.html>

<http://forum.runtimedna.com/showthread.php?19891-NURBs-versus-Spline-based-modelling&p=204889>

<http://www.integrityware.com/subdnurbs.html>

http://lightwiki.com/wiki/Fundamentals_of_Subpatch_Modeling

<http://forums.cgsociety.org/archive/index.php/t-575897.html>

<http://www.digitaltutors.com/forum/showthread.php?19078-What-is-the-difference-between-NURBS-subdivisions-and-polygons>

Técnicas de modelado

[http://www.3dbuzz.com/forum/threads/34439-Preference-of-modelling-Box-vs-Edge-](http://www.3dbuzz.com/forum/threads/34439-Preference-of-modelling-Box-vs-Edge-Extrude)

Extrude

<http://www.polycount.com/forum/archive/index.php/t-94272.html>

<http://www.polycount.com/forum/showthread.php?t=43090>

<http://theorangeduck.com/page/subdivision-modelling>

<http://3d.about.com/od/Creating-3D-The-CG-Pipeline/tp/Polygonal-3d-Modeling-Common-Box-And-Edge-Modeling-Workflows.htm>

Topología general

<http://forums.newtek.com/showthread.php?98754-Polygon-Table-Help-Building-One&s=74c8c9f7f0b57854037c92f70a715455>

<http://www.gotwires.info/search/label/basic>

<http://forums.cgsociety.org/showthread.php?t=764151>

<http://probiner.x10.mx/wiki/doku.php>

<http://www.foro3d.com/f122/edge-loop-concepto-practica-33375.html>

<http://www.polycount.com/forum/showthread.php?t=56014>

<http://wiki.polycount.com/SubdivisionSurfaceModeling>

<http://cgcookie.com/blender/2012/11/28/why-topology-matters-modeling/>

<https://vimeo.com/2158706>

http://chrisevans3d.com/tutorials/adv_subd.htm

<http://3dwisdoms.blogspot.com.es/>

Topología de la cara

<http://www.3dworldmag.com/2011/05/03/model-a-head-with-good-topology/>

<http://www.3dartistonline.com/news/2013/06/topogun-tips-for-better-topology/>

<http://lounge.ego-farms.com/showthread.php?t=213>

<http://www.gameartisans.org/forums/threads/24099-Propor-facial-topology-basics>

<http://www.foro3d.com/f112/modelado-texturizado-cabeza-utilizando-3ds-max-photoshop-61794.html>

<http://www.gameartisans.org/forums/threads/9196-Nevermore-Enters-The-Fray>.
<https://vimeo.com/66493749>

http://www.phungdinhdung.org/Studies_paper/Realistic_face_modeling.shtm

<http://cgcookie.com/blender/cgc-series/learning-mesh-topology-collection/>
http://www.youtube.com/watch?v=HZvhZ1OwuFk&feature=player_embedded#t=2
<http://www.gotwires.info/2013/07/video-tutorial-duylinh-nguyen-how-to.html>
http://www.sergicaballer.com/facial-modeling-timelapse_01/
<http://www.gotwires.info/2013/03/video-tutorial-wayne-robson-retopology.html>

Topología de todo el cuerpo

http://www.escultopintura.com.br/Tutoriais/Tutorial_Modelagem_Corpo_Humano/Sumario/Sumario_Corpo_Humano.htm
<http://cal.jmu.edu/ratner/tutorials/1/>

Tutoriales adicionales

<http://www.3dtotal.com/ffa/tutorials/max/joanofarc/joanmenu.php>

Foros completos

<http://web.archive.org/web/20101231110647/http://www.subdivisionmodeling.com/forums/index.php>

<http://www.polycount.com/forum/>
<http://wiki.polycount.com/>
<http://lightwiki.com/forums/viewtopic.php?f=4&t=22>
<http://www.gotwires.info/>
<http://www.gameartisans.org/forums/forum.php>